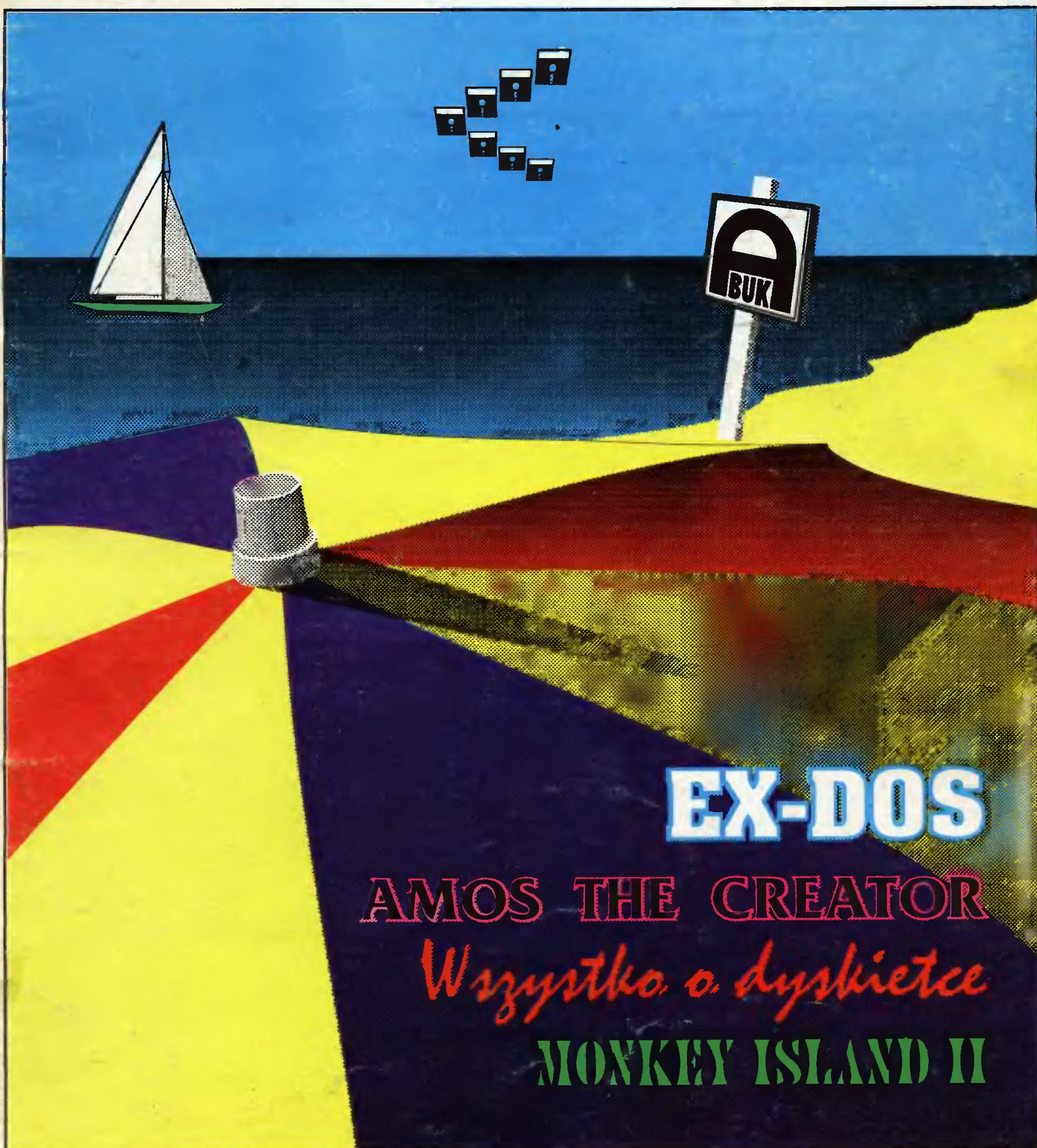
CENA 20.000 zł

ISSN 0867-3918 INDEKS 377112

64 PLUS 4 & AMIGA

7-8/'92

MIESIĘCZNIK UŻYTKOWNIKÓW KOMPUTERÓW COMMODORE



Informujemy, że nasze pismo można w dalszym ciągu **zaprenumerować** - co daje pewność systematycznego otrzymywania (drogą pocztową). Nasz miesięcznik kosztuje w prenumeracie 10.000 zł. Prenumeratę można zawrzeć na okres nie krótszy niż dwa miesiące, w dowolnym okresie, maksymalnie do końca roku kalendarzowego. Wykupujący prenumeratę nie ponoszą kosztów przesyłki pocztowej.

OD REDAKCJI



miesięcznik nr 7,8(21)
lipiec/sierpień 1992
cena 1 egz.: 10.000 zł.
64PLUS4

WYDAWCA: ABUK Spółka z o.o.
REDAGUJĄ: Waldemar Szczygieł (redaktor naczelny) z zespołem.
ADRES REDAKCJI: Redakcja „64 plus 4", 85-166 Bydgoszcz 43, skrytka pocztowa 64.
OKŁADKA: Piotr Bartz.
SKŁAD: ABUK
DRUK: W.Z.G. Wąbrzeźno

W numerze :

NOWINKI

Z DALEKA i Z BLISKA

PARTY

W dniach 27 oraz 28 czerwca w Żywcu odbył się kolejny zlot użytkowników komputerów Amiga. Spotkanie odbywało się w eleganckich salach miejscowego klubu.

Dużą atrakcją było wykorzystanie sali kinowej, w której na wielkim ekranie na początku można było oglądać filmy, a podczas konkursów pokazywane na nim były wszystkie obrazki oraz dema. Jak więc widać organizatorzy party nie zapomnieli o niczym. Już na stacji w Żywcu przez megafony uczestnicy spotkania zostali poinformowani o możliwych sposobach dojazdu na miejsce. Fajną sprawą było to, iż impreza zaczęła się zaraz po przyjeździe pierwszych uczestników, a nie dopiero o określonej godzinie, tak więc nie trzeba było czekać na wejście.

Do około godziny 14.00 na dwóch salach, w których odbywało się spotkanie miało miejsce totalne kopiowanie wszystkiego, co tylko się dało (tzw. swapping...). Po tej właśnie godzinie - na sali kinowej rozpoczął się konkurs na najlepszy moduł muzyczny. W tym czasie niestety zamknięte były dwie główne sale i wszyscy skazani byli na dwugodzinne, dość męczące przesłuchiwanie wszystkich modułów. Po skończonym konkursie muzycznym była chwila przerwy, po której miały miejsce kolejno konkursy na najlepszą grafikę oraz najlepszy program demonstracyjny.

W konkursie grafikcznym startowało dość dużo osób, natomiast programy demonstracyjne były tylko cztery. Wyniki były następujące:

Muzyka:
1. XTD/Action Direct
2. Roy B. Kyan/Future Revolution
3. Peters/Joker

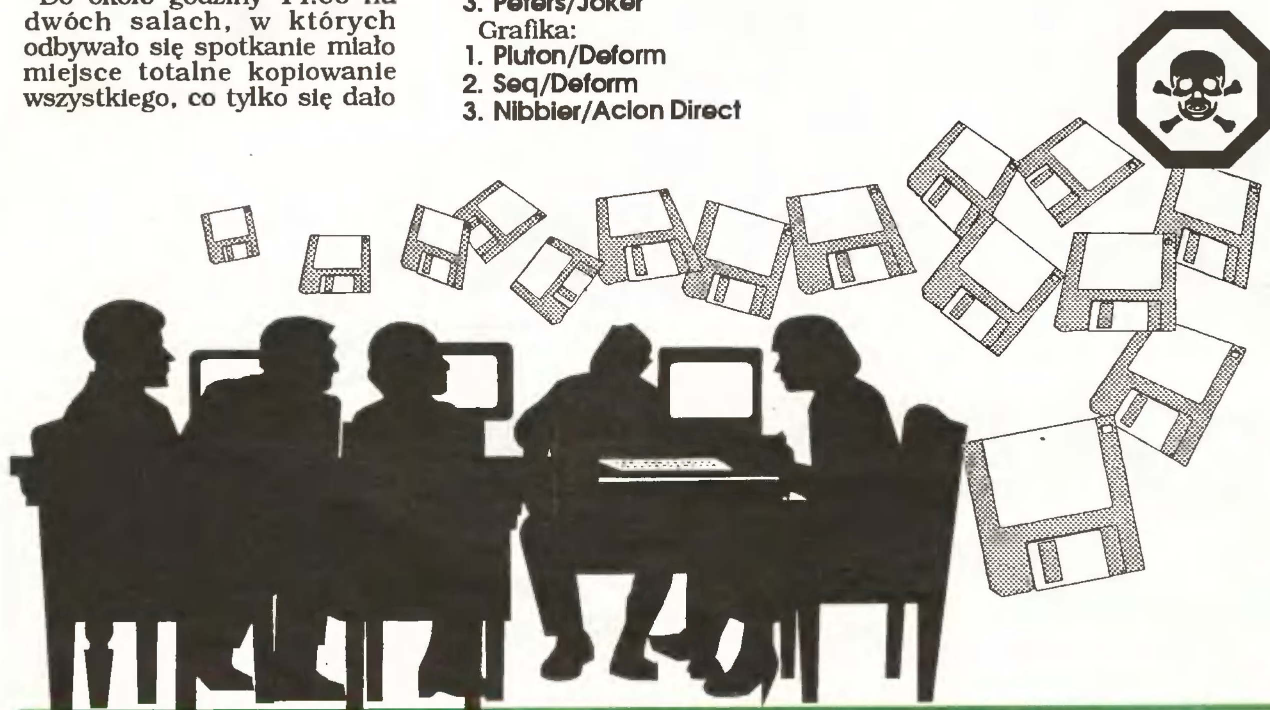
Grafika:
1. Pluton/Deform
2. Seq/Deform
3. Nibbier/Acion Direct

Demo:
1. "Immortal Visions" Suspect
2. "Damage" Old Bulls
3. "Hallucinations and Dreams" Katharsis

Niestety dema nie były najlepszej jakości w porównaniu z produkcjami prezentowanymi w Warszawie. Jednakże organizacja samych konkursów, za wyjątkiem kilku nieporozumień, była dużo lepsza, a zorganizowanie ich na sali kinowej z pewnością było bardzo dobrym rozwiązaniem.

Po skończonych konkursach sale ze sprzętem zostały ponownie otwarte i ponownie zaczęło się totalne przegrywanie wszystkiego co się dało. Podsumowując uważam, że organizacja party była najlepsza z dotychczas organizowanych tego rodzaju imprez, jednak prezentowane produkcje nie były najlepszej jakości.

Jarosław "Jarri" Horodecki



C-16

COLOUR BARS



Z pewnością każdemu zdarzyło się już spotkać, w którejś z gier czy programów demonstracyjnych, z efektownymi, kolorowymi paskami poruszającymi się po ekranie.

COLOUR BARS, bo o nich mowa, są szalenie zabawnym efektem wizualnym, który wykorzystać można pisząc własne programy, aby np. urozmaicić planszę tytułową, w nietypowy sposób wyczyścić zawartość ekranu itp.

Do takich właśnie osób, które kontakty z komputerem nie ograniczają jedynie do gier i chciałyby wiedzieć, w jaki sposób C+4 (16/116) może owe BARY generować, skierowany jest poniższy program.

Podstawą, od której należy rozpocząć, jest LISTING NR 1. Stanowi on właściwą procedurę wywołującą BARY, w tym przypadku, o szerokości 48 linii rastra (6 linii tekstowych). Zanim przystąpię do opisu możliwości programu, gwoli ścisłości dodam, że program ten jest w dużym stopniu przeróbką podobnego programu przeznaczonego na komputer C64 i przedstawionego przez pana Bartłomieja Kachniarza (B. Kachniarz, O przerwaniach rastra ciąg dalszy, Commodore & Amiga, 3(1992), s.27).

Po wpisaniu LISTINGU NR 1 i uruchomieniu go (G 1000), na ekranie powinien się pojawić nieruchomy barwny pas.

Aby dokładnie zapoznać się z działaniem programu, proponuję trochę poeksperymentować, przede wszystkim z zawartością komórki $1038 (do MONITORA przechodzimy naciskając RUN/STOP i RESET), która decyduje o tym, w którym miejscu na ekranie będzie wyświetlany BAR. Stąd już tylko krok do poruszania pasem, do czego służyć będzie LISTING NR 2, ale o tym za chwilę.

Kolory modyfikować można zmieniając wartość w tabelj kolorów ($106F - $109E).

Tabela pętli opóźnień ($103F - $106E) służy do wyrównywania długości poszczególnych linii (warto sprawdzić, co stanie się po zmianie wartości w tej tabeli; można pokusić się także o wyrównanie pasków, jeśli w przedstawionej formie wynik nie będzie jeszcze zadawalający). O ilości linii, z których składa się BAR, decyduje zawartość komórki $ 1034. Zmieniając ją (szczególnie wtedy, gdy chcemy rozszerzyć pas) należy pamiętać o modyfikacji tabeli opóźnień i tabeli kolorów (liczba wartości w tych tabelach musi zgadzać się z liczbą linii, które chcemy wyświetlić). Odczytem wartości z obu tabel zajmuje się część programu zawarta w komórkach $1023 - $1028.

LISTING NR 2 jest jedynie propozycją uzyskania animacji BARA na ekranie, dlatego nie będę szczegółowo objaśniał jego działania.

Aby połączyć oba programy, po wpisaniu LISTINGU NR 2, należy adres zapętlający w linii $1011 (JMP $1011) ustawić na początek procedury poruszającej BAREM (JMP $10AC).

Ruch odbywa się do dołu ekranu, po czym następuje odwrócenie kolejności kolorów, ruch do góry, znów odwrócenie kolorów itd.

Program 2 zmienia w programie 1 zawartość komórek $1038 (ruch), $1034 (zwężanie BARA przy odwracaniu kolorów) oraz $1022 i $1032 (aby układ odczytywał wartość z tabeli kolorów od końca - efektem jest odwrócenie kolejności wyświetlania linii BARA).

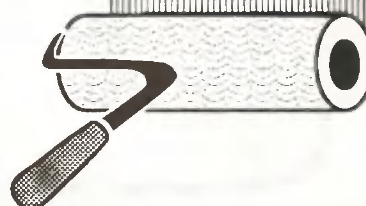
Podprogram w komórkach $109F - $10AB stanowi pętlę spowalniającą ruch. Prędkość ustalana jest w komórce $10A0.

Pole do eksperymentów jest duże. Znając podstawy programowania w języku maszynowym, bez trudu można napisać własny program animacji, uzyskać kilka niezależnie poruszających się BARÓW itp.

Uwaga: Tabela opóźnień i kolorów należy do LISTINGU NR 1 i razem z nim należy obie tabele wprowadzić do pamięci, aby program działał poprawnie.

LISTING NR 1:

```
1000  SEI          ;
1001  LDA #$14     ; Ustawienie IRQ
1003  STA $0314    ; na adres
1006  LDA #$10     ; $1014
1008  STA $0315    ;
100B  LDAQ #$02    ; Inicjalizacja przerwań
100D  STA $FF0A    ; rastra
1010  CLI          ;
1011  JMP $1011    ; Zapętlenie programu
1014  LDA #$02     ;
1016  STA $FF09    ; Potwierdzenie przerwań
```



C-16

```
1019    LDA $FF1D     ; rastra
101C    CMP $FF1D     ;
101F    BEQ $101C     ;
1021    LDX #$00      ; Wyświetlanie BARA
1023    LDA $106F,X   ; Odczyt kolejnej wartości
                        z tabeli kolorów
1026    LDY $103F,X   ; Odczyt kolejnej wartości
                        z tabeli opóźnień
1029    DEY           ;
102A    BNE $1029     ; Pętla opóźniająca
102C    STA $FF15     ; Wstawienie wartości do koloru
102F    STA $FF19     ; tła ekranu i ramki
1032    INX           ; Następna linia BARA
1033    CPX #$30      ; Sprawdzanie, czy wyświetlono
1035    BNE $1023     ; już 48 linii
1037    LDA #$30      ; Ustawienie przerwania rastra
1039    STA $FF0B     ; na linię $30
103C    JMP $CE0E     ; Wyjście z obsługi przerwania
```

TABELA OPÓŹNIEŃ:

>	103F	07	05	09	09	08	08	0B	06
>	1047	01	01	08	08	09	08	08	09
>	104F	02	01	08	08	08	08	08	08
>	1057	01	01	08	08	09	08	09	08
>	105F	01	01	08	08	08	09	09	08
>	1067	01	01	08	08	08	08	0A	09

TABELA KOLORÓW:

>	106F	F0	05	15	25	35	45	55	65
>	1077	75	65	55	45	35	25	15	05
>	107F	02	12	22	32	42	52	62	72
>	1087	72	62	52	42	32	22	12	02
>	108F	06	16	26	36	46	56	66	76
>	1097	66	56	46	36	26	16	06	F0

LISTING NR 2

```
109F    LDX #$05
10A1    CLC
10A2    LDA #$00
10A4    ADC #$01
10A6    BCC $10A4
10A8    DEX
10A9    BNE $10A1
10AB    RTS
10AC    LDA $1038
10AF    CMP #$D0
10B1    BEQ $10BC
10B3    INC $1038
10B6    JSR $109F
10B9    JMP $10AC
10BC    LDY #$FF
10BE    STY $1034
10C1    LDY #$2F
10C3    STY $1022
10C6    LDY #$CA
10C8    STY $1032
10CB    LDA $1038
10CE    CMP #$30
10D0    BEQ $10DB
10D2    DEC $1038
10D5    JSR $109F
10D8    JMP $10CB
10DB    LDY #$2E
10DD    STY $1022
10E0    DEC $1038
10E3    JSR $109F
10E6    DEY
10E7    BNE $10DD
10E9    NOP
10EA    LDY #$E8
10EC    STY $1032
10EF    LDY #$30
10F1    STY $1034
10F4    DEY
10F5    STY $1022
10F8    NOP
10F9    INC $1038
10FC    STY $1022
10FF    JSR $109F
1102    DEY
1103    CPY #$FF
1105    BNE $10F9
1107    JMP $10AC
```

Andrzej T. Witbrot

W poprzednim odcinku dokładnie omówiłem zasadę rysowania pojedyńczego punktu na ekranie. Dzisiaj proponuję zapoznać się z algorytmami tworzenia odcinka oraz wykorzystaniem tych algorytmów w praktyce.

C-64

KĄCIK POCZĄTKUJĄCEGO GRAFIKA

Na początek - dla przypomnienia - prezentujemy procedury niezbędne do korzystania z przykładów przedstawianych w tym artykule. Dokładniejsze ich omówienie zostało zaprezentowane w poprzednim artykule:

```
 999 REM Inicjacja grafiki wysokiej rozdzielczosci
1000 POKE 53272, PEEK (53272) OR 8
1010 POKE 53265, PEEK (53265) OR 32
1020 RETURN
1029 REM Przywrocenie trybu tekstowego
1030 POKE 53272, PEEK (53272) AND NOT 8
1040 POKE 53265, PEEK (53265) AND NOT 32
1050 RETURN
1059 REM Kasowanie ekranu graficznego
1060 FOR A=0 TO 999: POKE 1024+A,16*1+0: NEXT A
1070 FOR A=0 TO 7999: POKE 8192+A,0: NEXT A
1080 RETURN
1089 REM Rysowanie punktu na ekranie (X:Y)
1090 ADRES=8192+INT(Y/8)*320+INT(X/8)*8+(Y-
     INT(Y/8)*8)
1100 POKE ADRES, PEEK (ADRES) OR 2^(7-(X-INT(X/8)*8))
1110 RETURN
```

Teraz możemy się już zająć rysowaniem odcinka. Jak wiadomo odcinek jest to zbiór punktów należących do prostej. Tak więc, aby narysować go na ekranie wystarczy znaleźć podzbiór zbioru punktów prostej, w której się on zawiera. Prostą można przedstawić za pomocą równania: Y = M * X + B gdzie X oraz Y są punktami spełniającymi równanie tej prostej, M jest współczynnikiem nachylenia prostej względem osi układu współrzędnych, a B jest przesunięciem danej prostej. Można przyjąć, że B to wartość Y dla X=0. To tyle teorii.

Korzystając z tych wiadomości spróbujmy napisać program rysujący na ekranie prostą o zadanych współczynnikach M oraz B. Jak wiadomo punkt o współrzędnych (0;0) znajduje się na ekranie w jego lewym, górnym rogu, tak więc dla M i B ujemnego prosta ta znajdzie się poza obszarem widocznym na ekranie. To samo stanie się, gdy wartość M będzie dodatnia, a wartość B ujemna. Po przyjęciu tych dwóch podstawowych warunków musimy jeszcze uwzględnić kilka dodatkowych ograniczeń wynikających z wykorzystywanej przez nas rozdzielczości. W naszym przypadku mamy do dyspozycji 320 punktów w poziomie oraz 200 punktów w pionie (odpowiednio osie X oraz Y). Aby teraz móc określić położenie prostej na ekranie musimy porównać współrzędne punktów prostej z rozmiarami ekranu. Jeżeli współczynnik M naszej prostej jest ujemny, a B jest większe od maksymalnych rozmiarów ekranu w osi Y to skrajny lewy punkt tej prostej będzie położony na dolnym brzegu ekranu (wtedy Xpoczątkowe=(200-B)/M), przy czym może się zdarzyć, iż Xpoczątkowe przekroczy rozmiary ekranu i prosta nie będzie widoczna. Natomiast jeżeli B mieści się w rozmiarach ekranu to skrajny lewy punkt leży na lewym brzegu ekranu (Xpoczątkowe=0).

Ostatnią z możliwości jest M dodatnie oraz B ujemne. Wtedy skrajny lewy punkt leży na górnej krawędzi ekranu (i Xpoczątkowe=-B/M). Należy oczywiście sprawdzić, czy Xpoczątkowe mieści się w ramach ekranu. Jeżeli tak, to możemy rozpocząć w końcu rysowanie odpowiedniego fragmentu naszej prostej. W oparciu o podane warunki łatwo jest napisać program rysujący prostą mając zadane parametry M oraz B. Oto moja propozycja takiego programu:

```
 10 PRINT CHR$(147);
 20 INPUT "PODAJ NACHYLENIE
    PROSTEJ: "M
 30 INPUT "PODAJ PRZESUNIECIE
    PROSTEJ: "B
 40 IF (M<0) AND (B<0) THEN END
 50 IF (M>0) AND (B>200) THEN END
 60 XP=0
 70 IF (M<0) AND (B>200) THEN
    X1=(200-B)/M: IF X>320 THEN END
 80 IF (M>0) AND (B<0) THEN
    X1=-B/M: IF X>320 THEN END
 90 GOSUB 1000
100 GOSUB 1060
110 FOR X=XP TO 320
120 Y=M*X+B
130 IF (Y<0) OR (Y>200) THEN END
140 GOSUB 1090
150 NEXT X
160 END
```

Oczywiście progam ten jest jedynie wstępem do rysowania odcinków przy pomocy komputera.

Naszym drugim krokiem jest stworzenie programu rysującego odcinek o podanych współrzędnych początko-

wych oraz końcowych. Czy rzeczywiście jest to takie trudne? Ależ nie! Przyjmijmy, że odcinek ma początek w punkcie o współrzędnych (X1;Y1), a koniec w punkcie o współrzędnych (X2;Y2). Teraz wystarczy znaleźć wartość nachylenia (M) oraz przesunięcia (B) - wystarczy wykonać kilka bardzo prostych działań, mianowicie:

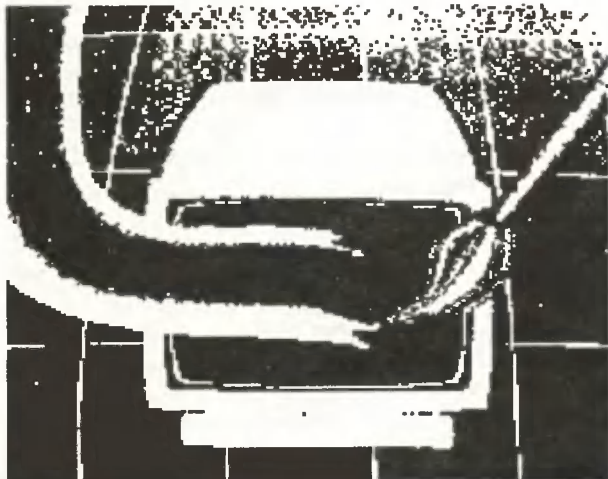
M = (Y2 - Y1) / (X2 - X1)

B = Y1 - M * X1

Ze względu na to, iż nie musimy już dbać o to, czy prosta znajduje się na ekranie (mamy dane współrzędne) możemy zlikwidować wszystkie warunki. Proponuję napisać nasz program już w formie procedury, którą "doczepimy" do naszego pakietu:

```
1119 REM Rysowanie odcinka od (X1:Y1) do (X2:Y2)
1120 M=(Y2-Y1)/(X2-X1): B=Y1-M*X1
1130 FOR X=X1 TO X2
1140 Y=M*X+B: IF (Y<0) OR (Y>200) THEN 1160
1150 GOSUB 1090: NEXT X
1160 RETURN
```

Tak to będzie wyglądać nasza procedura rysowania linii. Niestety ma on kilka, dość istotnych wad. Po pierwsze jest bardzo wolna (jest to jednak wada każdego programu napisanego w Basicu). Drugą wadą jest niedokładność jej działania. Linia nie jest ciągła, ale składa się z kolejnych punktów. Trzecią, dużą wadą, jest konieczność podawania zawsze X1 mniejszego od X2. W przeciwnym wypadku na ekranie pojawi się tylko jeden punkt. Można oczywiście procedurę tą udoskonalać, jednak pozastawię to już czytelnikom jako ewentualne ćwiczenie.



My natomiast zastosujemy teraz całkiem inny algorytm, pozbawiony dwóch z wymienionych wad. Po pierwsze współrzędne można już podawać zupełnie dowolnie, a po drugie rysowane odcinki są spójne i wyglądają lepiej. Trzecia wada jednak ciągle pozostaje, a jedyną metodą jej usunięcia jest zastosowanie języka maszynowego, na co mamy jeszcze trochę czasu. Przejdźmy jednak do naszej nowej procedury.

Skasujmy poprzedni podprogram rysowania linii i w jego miejsce wpiszmy ten oto progamik:

```
1120 XA=0: YA=0
1130 N=ABS(X2-X1): M=ABS(Y1-Y2)
1140 IF N<M THEN N=M
1150 IF N=0 THEN END
1160 DX=(X2-X1)/N: DY=(Y2-Y1)/N
1170 FOR I=1 TO N
1180 X=INT (X1+XA)
1190 Y=INT (X1-XA)
1200 GOSUB 1090
1210 XA=XA+DX: YA=YA+DY
1220 NEXT I
1230 RETURN
```

Podprogram ten może się wydawać nieco bardziej skomplikowany, jednakże w rzeczywistości jego zasada działania jest prosta. Algorytm polega na określeniu współrzędnej, według której będziemy rysować odcinek. Za zmienną sterującą pętli przyjmowana jest ta współrzędna, która wymaga większej ilości kroków. Po jej wyznaczeniu obliczany jest krok, z jakim mają przyrastać obie współrzędne, przy czym zawsze jedna z nich będzie się zwiększać o jeden. Po wykonaniu każdego kroku rysowany jest jeden punkt, aż do otrzymania całego odcinka.

W ten sposób mamy już zestaw podstawowych procedur do tworzenia grafiki wysokiej rozdzielczości. W następnym odcinku poznamy sposób rysowania okręgu oraz spróbujemy przedstawić na ekranie wykresy różnych funkcji matematycznych.

Jarosław "Jarri" Horodecki

„MODEMOWANIE" cz. 3

C-64

Po zapoznaniu się z podstawowymi wiadomościami z dziedziny telekomunikacji poznamy dziś, w jaki sposób człowiek za pośrednictwem komputera może porozumiewać się z modemem. Zapoznamy się więc z komendami oraz z raportami modemów.

Zapewne wszyscy wiedzą, że modem jest urządzeniem, które korzysta z portu szeregowego naszego komputera. Posiadacze Amigi, Atari ST, PC czy innych "dużych" maszyn nie będą mieli z tym kłopotu, gdyż takie interface'y są już w te komputery wbudowane. Jednak posiadacze C64 będą musieli jeszcze zakupić specjalną kartę wkładaną do user-portu, która zawiera gniazdo interface'u RS-232. Drugą dość istoną sprawą jest program, którym możemy się posługiwać w pracy z modemem. Tu również posiadacze tej pierwszej grupy komputerów nie będą mieli kłopotów, gdyż istnieje bardzo wiele programów mogących kontrolować pracę praktycznie każdego modemu. Natomiast w komputerach typu C64 po pierwsze maksymalna prędkość portu szeregowego wynosi 2400 bodów, a po drugie programy terminali nie są zbyt wygodne i (z wyjątkiem nielicznych programów) nie mają możliwości wygodnej edycji listy telefonów i wielu innych przydatnych opcji. Dość istoną sprawą jest także zastąpienie w C64 znaczka "\" znakiem funta.

Po tym krótkim wstępnie przejdźmy do zasadniczej części artykułu, czyli omówienia komend. Posłużymy się przykładem modemu 2400 bodów wyposażonego w protokół MNP5, czyli najpopularniejszego obecnie rodzaju modemów w Polsce. Zacznijmy od tego, iż każda komenda (z wyjątkiem dwóch) zaczyna się od liter "AT". Litery to pochodzą od słowa "attention" (uwaga). Wpisanie ich daje sygnał modemowi, że za chwile będzie wydana komenda, a nie strumień danych. Tak więc zawsze, przed każdą komedną musimy najpierw wpisać te dwie litery i dopiero wtedy, bez naciskania RETURN literę odpowiadającą danej komendzie. I tak na przykład nakazanie modemowi, aby podniósł słuchawkę i nadał sygnał będzie wyglądałą tak:

ATA.

AT to nasz przedrostek, a A to symbol tej właśnie komendy.

Oto spis wszystkich standardowych komend:

B0 - wybranie trybu pracy zgodnego z normą V.22/V.22bis dla prędkości 1200 oraz 2400 bodów lub V.21 dla 300 bodów.
B1 - tryb pracy Bell 103/212A dla prędkości 300 oraz 1200 bodów.
B2 - tryb pracy V.23 dla prędkości.
D - wybieranie numeru (poźniej omówione szerzej).
E0 - wyłączenie echa, tzn. znaki wysyłane do modemu nie są odsyłane do terminala, czyli nie widać tego, co się pisze.
E1 - włączenie echa.
H - odłożenie słuchawki.
I0 - podanie kodu identyfikacyjnego modemu.
I1 - podanie sumy kontrolnej firmowego ROMu.
I2 - sprawdzenie prawidłowości zawartości firmowego ROMu.
I3 - podanie kodu identyfikacyjnego MNP5.
L0 - mała głośność głośnika w modemie.
L1 - mała głośność głośnika w modemie.
L2 - śrenia głośność głośnika w modemie.
L2 - duża głośność głośnika w modemie.
M0 - głośnik zawsze wyłączony.
M1 - głośnik włączony aż do złapania sygnału nośnego.
M2 - głośnik zawsze włączony.
M3 - głośnik zawsze włączony aż do złapania sygnału nośnego, ale wyłączony podczas wykręcania numeru.
O0 - wejście w tryb przesyłania danych.
O1 - wejście w tryb przesyłania danych z automatycznym dostrojeniem się do warunków transmisji.
V0 - kody wynikowe podawane w formie liczbowej.
V1 - kody wynikowe podawane w formie znakowej.
X0 - zgodność z modemem Hayes Smartmodem 300.
X1 - wyświetlanie raportów CONNECT 1200 oraz CONNECT 2400.
X2 - włączenie wykrywania sygnału centrali.
X3 - włączenie wykrywania sygnału zajętości.
X4 - właczenie wykrywania zarówno sygnału centrali, jak i sygnału zajętości.
Y0 - modem nie wysyła sygnału przerwania.
Y1 - modem wysyła sygnał przerwania przez 4 sekundy przed rozłączeniem.
Z - reset modemu i ustawienie standardowych parametrów.

Komenda "D", czyli wykręcanie numeru wymaga dokładniejszego omówienia. Można po niej stosować dodatkowe komendy informujące o sposobie wykręcania numeru, i tak:

P - wybieranie impulsowe (stosowane w Polsce).
T - wybieranie tonowe (nie stosowane w Polsce).
, - krótka przerwa w wybieraniu.
@ - czekanie na ciszę.
w - czekanie na sygnał centrali.
! - przerwa długości 0,5s.
; - po wybraniu numeru powrót do trybu przyjmowania komend.

W zależności od modelu mogą jeszcze istnieć inne komendy.

Oprócz komend, bardzo przydatna jest także znajomość rejestrów modemu, w których przechowywanych jest bardzo wiele istonych informacji. Przypisanie wartości danemu rejestrowi odbywa się przez wydanie komendy

Sxx=nn

gdzie xx jest numerem rejestru, a nn wartością, jaką danemu rejesrowi chcemy przypisać.

Oto spis wszystkich podstawowych rejestrów:

S00 - po ilu dzwonkach modem ma podnieść słuchawkę (od 0 do 255), przy czym podanie wartości 0 wyłącza automatycznie odpowiadanie.

C-64

S01 - ile dzwonków czekać na odpowiedź (0-255).
S02 - kod znaku wyjścia (ESCape), standardowo 43.
S03 - kod znaku CR, standardowo 13.
S04 - kod znaku przejścia do następnej linii, standardowo 10.
S05 - kod znaku kasowania (BACKSPACE), standardowo 8.
S06 - czas oczekiwania na sygnał centrali (2-255 sekund).
S07 - czas czekania na sygnał nośny (1-255 sekund).
S08 - długość pauzy w wybieraniu wywołanej przecinkiem (0-255 sekund).
S09 - czas odpowiedzi na sygnał nośny (1-255 1/10 sekundy).
S10 - czas pomiędzy zgubieniem sygnału nośnego, a przerwaniem transmisji (1-255-1/10 sekundy).
S11 - przerwy pomiędzy sygnałami wybierania tonowego (50-255 ms).
S12 - czas, po którym zadziała kod wyjścia (20-255 1/50 sekundy).
S16 - testy modemu.
S18 - czas testu (0-255 sekund).

Pozostałe rejstry oraz te pominięte są bądź zarezerwowane, bądź też są rejestrami flagowymi modemu, na których stan można wpływać za pośrednictwem zwykłych komend.

Oprócz komend, które wydajemy modemowi warto poznać jeszcze raporty, jakie od modemu otrzymujemy. Oto ich spis wraz z krótkimi opisami:

OK - standardowa odpowiedź, po przyjęciu i zrealizowaniu wydanego polecenia.
ERROR - podanie nieznanego rozkazu.
CONNECT - połączenie z prędkością 300 bodów.
CONNECT 1200 - połączenie z prędkością 1200 bodów.
CONNECT 2400 - połączenie z prędkością 2400 bodów.
RING - modem przyjmuje dzwonek.
NO CARRIER - podczas transmisji został zgubiony sygnał nośny.
NO DIALTONE - nie został wykryty sygnał centrali.
BUSY - został wykryty sygnał zajętości.
NO ANSWER - nie została wykryta cisza (tylko komenda @).

Na koniec jeszcze rozkazy nie wymagające przedrostka AT. Istnieją tylko dwa takie rozkazy. Pierwszy z nich to "A/", który powoduje powtórzenie ostanio wykonywanego polecenia, a drugi to "+++", który powoduje przejście modemu z trybu transmisji danych do trybu przyjmowania rozkazów.

To już wszystko, jeżeli chodzi o podstawowe wiadomości na temat komend, którymi sterujemy modemem. Oprócz nich najczęściej istnieje szereg innych rozkazów rozpoczynających się od takich znaczków jak "&", czy "%", jednak są one zależne od producenta modemu i dokładne informacje o nich zamieszczane są w instrukcjach obsługi konkretnego modemu.

Jarosław "Jarri" Horodecki

OGŁOSZENIA

Nawiążę kontakt w celu wymiany oprogramowania C+4 P. Rybok BĘDZIN ul. Kniewskiego 1/77.

Komputerowa firma usługowa "TREND" Commodore Amiga 500 - 3000 literatura w j. polskim (!) i oprogramowanie Informacja: dyskietka lub koperta + znaczek. Kontakt: Rafał Wierzbicki, ul. Budziszyńska 112/28, 54-436 Wrocław.

SPRZEDAM oryginalny programowy emulator IBM'a (CGA) - 50.000 zł PIOTR LASZCZYK, 34-511 KOŚCIELISKO, SZELIGÓWKA 976.

Sprzedam 1-o roczną Amigę 500 z 1MB pamięci za 5.000.000 zł Ciepłucha Radosław 11-go Listopada 71/48 91-372 Łódź.

Mapa pamięci do C-64 po polsku, wiadomość: koperta + znaczek skr. poczt. 39 11-600 Węgorzewo. Zawsze aktualne!

THE MANY AMI nawiąże kontakt z Amigowcami info k. + zn. Kuba Kopczyk Os. Wichrowe Wzgórze 26/73 61-697 Poznań

Nowo założona grupa "TMS" poszukuje zdolnego rysownika ołówkiem. Adres: ul. 1 Maja 4 88-230 Piotrków Kuj. telefon 286.

Poszukuję programu do nauki niemieckiego (C-64, kaseta) zwrócę nośnik. Artur Drążyk ul. Pułaskiego 10 m 10 05-400 Otwock.

Kupię stację 1581 i moduł 1750 M. Sokołowski, ul. Kazimierza Wlk. 28/38 Kielce, tel. 541-74

Sprzedam C-64 II + magnetofon, jojstik, black box 4.0, 4 kasety (gwarancja) za 2.500.000 zł lub zamienię na PC/XT z dopłatą 500.000 zł. Piotr Murdzek 59-300 Lublin ul. Sokola 45/15 tel. 44-56-87

Wymienię C-64 z magnetofonem, kasetami, 2 kartridge, 2 jojstiki, monitor PHILIPS, czasopisma, książki, instrukcje obsługi, kabel łączący monitor z klawiaturą na Amigę 500 i parę dysków. Sebastian Musiał Istebna 500 43-460 Wisła woj. Bielsko Biała

OPIS ZESTAWU PUBLIC DOMAIN PACK

nr 18 (dysk - czerwiec '92)

C-64

W końcu mamy przed sobą nową dyskietkę PDP. Zobaczmy co na niej się tym razem znajduje. Napewno znajdziemy tu jak zwykle wiele ciekawych programów demonstracyjnych oraz użytkowych. W czerwcowym PDP zamieszczamy teoretycznie tylko trzy programy, w tym jeden program demonstracyjny, jednen magazyn dyskowy oraz zbiór programów graficznych. Zaczniemy oczywiście od użytków.

GFX DESIGNERS PACK V1.0. Jest to nowa produkcja grupy TAT, która wszytkie najlepsze oraz najbardziej znane programy graficzne złożyła na jeden dysk i opatrzyła eleganckim menu oraz swoją czołówką. W pakiecie tym znajdują się wszytkie niezbędne każdemu grafikowi programy, począwszy od programów służących do rysowania, a skończywszy na obróbce gotowych rysunków. Kilka z zamieszczonych tu programów były już kiedyś publikowane w PDP, jednak ze względu na chęć zachowania całości pakietu nie kasowaliśmy ich. Poza tym są to zwykle krótsze ich wersje, a co za tym idzie nieco wygodniejsze w obsłudze. Proponuję zająć się teraz dokładniejszymi opisami.

1. Picture Shower V1.0. Najnowsza wersja dość prostego programiku służącego do efektownego przedstawiania prac grafików. Został napisany przez Scrappera z grupy Blaze. Program ten umożliwia wczytanie dowolnego obrazka zapisanego w formacie Koala Microilustrator. Konwersji na ten standard można dokonać przy pomocy niemalże każdego konwertera grafiki. Po załadowaniu do pamięci własnego obrazka, możemy wczytać jeszcze muzykę oraz dowolny krój czcionki o wielkości 8 na 8 pikseli. Następnie wystarczy napisać komentarz do naszego obrazka (ukaże się on przed jego pokazaniem) oraz tekst do scrolla, który będzie się przewijał pod obrazkiem. W końcu wystarczy sprawdzić, czy wszytko jest w porządku oraz spakować przygotowany w ten sposób obrazek. Teraz możemy się naszymi osiągnięciami pochwalić przed znajomymi...

2. Side-Border Logo Editor V1.0. Program graficzny do rysowania obrazków, które wychodzą na boczną ramkę. Efekt ten nie należy do najprostszych, więc nie będę teraz opisywał sposobu jego działania. Skupmy się na samym programie. Niestety, nie posiada on zbyt dużej ilości opcji. Umożliwia jedynie rysowanie podstawowych figur płaskich, (proste, okręgi). Program posiada też bardzo przydatną opcję kopiowania bloków. Po skończeniu rysunku możemy zgrać go jako normalny obrazek w trybie multicolor lub też jako obraz znakowy i odpowiedni generator znaków. Oczywiście do każdego z obrazków dopisywane są odpowiednie sprite'y odpowiadające fragmentom wychodzącym na ramkę boczną.

3. Add Animator V1.0. Program ten służy do transformacji obrazków. Aby móc z niego skorzystać najpierw należy wgrać rysunek przeznaczony do obróbki. Następnie, po wycięciu odpowiedniego jego fragmentu, określamy ilość faz animacji, w jakiej ma być dokonana transformacja oraz wielkość obrotu wokół każdej z osi. Po wybraniu tych opcji komputer zacznie rysować kolejne fazy animacji oraz zgrywać je na dysk. Po skończeniu pracy możemy wykorzystać przygotowane w ten sposob logo we własnym demie.

4. Fli Graph Editor V2.2. Ten program to jeden z najlepszych (jeżeli nie najlepszy) edytor do tworzenia grafiki w trybie FLI. Posiada ogromną liczbę opcji pomocnych w tworzeniu własnych obrazków. Wadą jest konieczność wybierania wszystkich opcji z klawiatury co niesie za sobą konieczność nauczenia się obsługi lub też częstego wywoływania dokładnej instrukcji. W odróżnieniu od większości innych programów tego rodzaju Fli-Graph daje także możliwość rysowania na trzech pierwszych kolumnach ekranu. Występują jednak w nich pewne ograniczenia w stosowaniu kolorów wynikające ze specyfiki trybu FLI.

5. Fli/Koala Converter. Prosty program służący do konwersji dowolnych obrazków z trybu FLI na Multi oraz odwrotnie. Po tej konwersji wymagane są zwykle tylko drobne poprawki kolorów w miejscach, gdzie FLI było naprawde potrzebne. Oczywiście obrazek stworzony przez dobrego grafika obeznanego z techniką FLI i umiejętnie ją wykorzystującego raczej nie będzie możliwy do zamiany na zwykły tryb multi.

6. Colour Editor. Program służący do edycji kolorów dowolnego rysunku. Poruszając się kursorem po jego powierzchni możemy dokonywać zmian kolorów dowolnych kwadratów o wymiarach 4 na 8 pikseli. W ten sposób możemy nasze dawne rysunki czterokolorowe przerobić na rysunki wykorzystujące pełne możliwości C64 i po takiej przeróbce korzystać z nich przy pisaniu naszego kolejnego programu demonstracyjnego.

7. Hires Menager. Produkt znanego programisty z Austrii - Hannesa Sommera. Jest to edytor służący do tworzenia grafiki w trybe A-FLI. Obecnie uznany jest za najlepszy z tego rodzaju edytorów. Jest bardzo wygodny oraz prosty w obsłudze. Mimo to posiada sporo interesujących i przydatnych opcji, np. rzadko spotykana w tego rodzaju programach funkcja UNDO. Program posiada także dość ciekawie rozwiązany sposób samego procesu rysowania obrazu.

8. Hires Converter. Jest to chyba najbardziej rozbudowany program służący do konwersji pomiędzy wszelkimi znanymi standardami zapisu obrazków. Możemy wczytywać obrazki z takich programów jak Art Studio, Advanced Art Studio, Artist 64, Blazzing Paddles, Cadmaster, Cdu Paint, Doodle, Image System, Paintbox 2001, Paint Magic, Vidcom 64. Jak widac biblioteka dostępnych standardów jest bardzo duża. Oprócz konwersji możemy także zamieniać obrazki na generator wraz odpowiednim obrazem znakowym lub też dokonywać zamiany generatora na obrazek w wybranym standardzie.

9. E.M.C. Paint. Jest to kolejny, program do tworzenia grafiki w technice FLI.

10. 64'8 Logo Editor. Program ten to jedyny znany mi edytor pozwalający na rysowania obrazków o długości większej niż jeden ekran. Dzięki możliwości przesuwania tworzonego rysunku możemy stworzyć logo o wysokości 8 wierszy i długości 64 kolumn. Autorzy programu zadbali o wpro-



C-64

wadzenie kilku ciekawych opcji np.: możliwość zapamiętania dwóch niezależnych rysunków na raz w pamięci. Wadą programu jest możliwość rysowania przy użyciu tylko czterech wybranych kolorów.

To już wszystkie programy użytkowe zamieszczone tym razem na PDP. Oczywiście nie zapomnieliśmy o fanach programów demonstracyjnych.

Dla nich to zamieściliśmy bardzo dobre demo znanej grupy Camelot pod tytułem "One Year Camelot II". Program ten zawiera bardzo wiele ciekawych i nowych efektów. Na początku obejrzeć możemy przyjemne intro z gwiazdkami popularnymi na Amidze. Kolejna część to elegancko wykonany sinus-scroll, ale... zajmujący cały ekran, a każda ze sporej wielkości liter wygina się co jeden piksel! Następna część to kolejny nowy efekt. Tym razem nieco łatwiejszy do wykonania, jednak obmyślenie takiej procedury wymagało od autorów dużej pomysłowości. Część ta to dość specyficznie wykonany efekt plazmy. Następnie możemy obejrzeć kule składającą się ze 128 kropek obracających się wokół trzech osi. Warto zaznaczyć, że wszystkie obliczenia obrotów wykonywane są w czasie rzeczywistym. Po kuli przyszedł czas na linie. I oto w następnej części widzimy 8 różnokolorowych linii poruszających się po różnych torach. Po naciśnięciu spacji przechodzimy do następnej fazy, w której autorzy dema prezentują nam wspaniałą wektorówkę. Jest to tak zwany TV-Box, czyli sześcian, na każdej ze ścian znajduje się inny, również ruchomy obiekt. Tutaj także obliczenia wykonywane są w czasie rzeczywistym, a prędkości obrotów możemy regulować za pomocą joysticka. Następna część dema nie jest może już aż tak rewelacyjna, jednak jest to kolejny pobity przez autorów rekord. Tym razem możemy obejrzeć 184 dość spore bobsy ruszające się według funkcji sinus. W kolejnej części zaprezentowano dość skomplikowany efekt wymagający pewnych umiejętności w dziedzinie tzw. cyklowania. Jest to multiplexer (więcej niż osiem spriteów) nałożony na 8 tzw. splitów, czyli linii w których zmieniamy kilkakrotnie kolor ekranu. Przedostatnia już część to następny rekord, tym razem w ilości scrolli. Na ekranie widzimy 15 całoekranowych sinus-scrolli poruszających się po różnych torach. Ostania część to efekt również ściągnięty z Amigi. Są to tak zwane pionowe rastry, czyli poruszające się w prawo oraz w lewo kolorowe, pionowe pasy.

Na koniec zamieściliśmy jeszcze jeden krótki programik: ADDY CORNER numer 4. Jest to zachodni magazyn dyskowy, w którym umieszczone są w zasadzie tylko adresy ludzi z całego świata chcących wymieniać się programami. Myślę, że wszyscy pragnący nawiązać kontakty z kimś z poza granic Polski znajdzie tam coś dla siebie.



I to już wszystkie programy zamieszczone na czerwcowym PDP. Mamy nadzieję, że okażą się one dla was przydatne i ciekawe.

Public Domain Pack

nr 19 (lipiec '92)

W tym numerze wyjątkowo opisujemy aż dwa kolejne Public Domain Packi. Lipcowy PDP zawiera kilka ciekawych programów użytkowych oraz demonstracyjnych. Na początek, jak zawsze, prezentujemy krótkie opisy wszytkich zamieszczonych na naszym PDP programów użytkowych.

Hack Pack V2.1 to zestaw najczęściej używanych i najbardziej efektywnych kompresorów. Zawarte są w nim następujące programy:

Cruel Cruncher V2.5 to jeden z najlepszych kompresorów sekwencyjno-bitowych. Może pracować, w zależności od wymagań, z jedną z ośmiu prędkości (od 1 do 7 oraz tzw. Hyperspeed), która determinuje stopień kompresji, chociaż w niektórych przypadkach może się zdarzyć, iż osiągniemy lepszy rezultat przy większej prędkości.

Drugim z kompresorów jest Fast Cruel V2.5. Jest on właściwie bardzo podobny do Cruel Crunchera, jednakże wykorzystuje dużo szybsze, lecz mniej efektywne algorytmy pakowania.

Następnym kompresorem jest Level Cruel V2.2. Jest to dość specyficzny rodzaj kompresora służący do pakowania plików bez dołączania procedury dekompresującej. Kompresory tego rodzaju służą w głównej mierze do kompresji danych do kolejnych poziomów nielegalnie kopiowanych gier, ale mogą być też wykorzystywane do kompresji kolejnych części programu demonstracyjnego, czy też doczytywanych przez jakiś program muzyczek lub obrazków.

Kolejnym kompresor to Zipper V5.0. Jest to dość efektywny kompresor typu znakowego. Może on pakować obszar pamięci od adresu $0790 aż do $FFFF.

Bardzo podobny do poprzedniego kompresora jest $54 Zipper. Dużą zaletą tegoż kompresora jest długość procedury dekompresującej. Liczy ona tylko $54 bajty! Kompresor może pakować dane z obszaru pamięci od $07E0 do $FFFF.

Ostatnim z programów w tym pakiecie jest FW4C packer. Może on skompresować niemalże całą zawartość pamięci, bo od adresu $0200 aż do $FFFF. Kompresja odbywa się metodą znakową (tzw. char-packer).

Ex-DOS and Disk Doctor to moim zdaniem jeden z najlepszych monitorów dyskowych dostępnych a C64. Ma on dość dużo przydatnych opcji, dlatego też jego omówieniu poświęcimy osobny artykuł w naszym cyklu struktura dysku. Tym samym omówienie tego programu będziemy mogli połączyć z poznaniem kilku nowych informacji zapisanych na dyskietce.

Fast Hack 'em V4.4a to najnowsza wersja dobrze znanego wszystkim użytkownikom C64 programu kopiującego. Tym razem program nie zajmuje jednak całej strony dyskietki, ale zaledwie kilkudziesięcioblokowy pojedyńczy plik, w którym zawarte są wszystkie najważniejsze narzędzia służące do kopiowania. Mamy oczywiście podstawowe kopiery do całych stron dyskietek działające zarówno z jedną jak i z dwoma stacjami dysków. Są także specjalne programy do kopiowania zabezpieczonych programów. Umożliwiają one nawet kopiowanie błędów, zgodnie z zawartością dysku źródłowego. Oczywiście w pakiecie znajduje się także bardzo dobry kopier do plików, także w dwóch wersjach:

dla jednej i dla dwóch stacji dysków. Wszystkie operacje można wykonywać z weryfikacją lub bez niej.

Jest tu także kilka innych ciekawych programików. Jednym z nich jest prosty, ale bardzo wygodny edytor sektorów dyskietki. Posiada on wszystkie podstawowe funkcje jak zapis i odczyt sektora oraz jego edycja za pomocą znaków lub też pojedyńczych bajtów.

Music Wizard V3.0 jest programem bardzo przydatnym dla wszystkich koderów. Służy do łatwego i wygodnego znajdywania w procedurach odtwarzających muzyczki bajtów odpowiedzialnych za wysokość, kształt fali oraz inne dane opisujące każdy z odgrywanych dźwięków. Sprawdzać to wszytko można podczas odtwarzania muzyczki, co jest bardzo wygodne, gdyż umożliwia przeglądanie na bieżąco wszystkich zmian zachodzących w procedurze. Program jest bardzo starannie wykonany i bardzo wygodny w obsłudze co stawia go na pierwszej pozycji wśród tego rodzaju narzędzi.

Digi Ripper V1.0 to dość prosty, a jednak bardzo wygodny programik służący do znajdowania w pamięci samplowanych dźwięków. Mogą to być zarówno pojedyńcze instrumenty z niektórych utworów jak również fragmenty utworów samplowane jako jedna całość (tzw. digitalizacje). Program umożliwia przeglądanie niemalże całego obszaru pamięci. Możemy przeglądać dane od adresu $2000 aż do końca pamięci. Każdy fragment przedstawiany jest także graficznie. Poza tym mamy oczywiście możliwość odtworzenia dowolnego fragmentu z dowolną prędkością oraz, co jest chyba najważniejsze, zgrania go na dyskietkę.

SCS Relocator V1.0 oraz JCH Relocator to dwa programy o bardzo podobnym działaniu. Służą one do zmieniania adresów całych utworów muzycznych napisanych przy użyciu jednej z dwóch procedur odtwarzających: grupy SCS lub znanego muzyka JCH.

Aby użyć tych programów należy najpierw upewnić się, iż utwór, którego adres mamy zamiar zmienić napewno jest zgodny z jedną z tych procedur. Następnie wystarczy już tylko wgrać potrzebny relokator, następnie nasz utwór przeznaczony do obróbki oraz podać programowi adres docelowy. Po chwili będziemy mieli już gotowy, przerelokowany kod wraz z danymi utworu.

Często się jednak zdarza, że w wyniku zaistanienia pewnych niezgodności - mimo nie wykrycia błędu przez relokator - muzyczka nie będzie działała. Niestety możemy wtedy albo użyć zwykłego relokatora do programów w języku maszynowym albo wykonać próby z inną muzyczką.

Message and Command Editor to świetny programik służący do prostej zmiany wszystkich raportów zawartych w pamięci ROM komputera. Możemy zmienić komunikaty o błędach oraz cały zestaw intrukji. Następnie zgrać tak zmienioną pamięć ROM oraz po zresetowaniu komputera wgrać ten plik przy pomocy komendy LOAD "nazwa",8,1 i wykonać instrukcję POKE 1,54 i możemy już korzystać z własnych komunikatów oraz komend. Należy jednak uważać, aby nie wykonywać rozkazów POKE od adresu 40960 do 49151 gdyż może to spowodować zawieszenie się komputera lub też błędne wykonywanie niektórych rozkazów (co też może doprowadzić do zawieszenia komputera).

Na dysku nagrana jest niemiecka oraz oryginalna angielska wersja pamięci ROM.

Deadline Writer V2.0 to bardzo prosty wyposażony jedynie w podstawowe funkcje edytor do pisania tekstów scrolli ("płynących napisów"). Opis wszytkich dostępnych funkcji jest pokazany zaraz po uruchomieniu programu.

Sledge Hammer V3.0 to najnowsza wersja bardzo dobrego kompresora znakowego. Może on pakować dane od adresu $0500 aż do $FFFF. Wersja ta posiada szybkie procedury transmisji danych z dyskiem oraz lepszy algorytm pakowania. Program ten umożliwia także wprowadzenie jednej linii tekstu, która ma być wyświetlana podczas dekompresji. Ciekawą i dość przydatną właściwością programu jest także możliwość łączenia kilku (do 24) plików. Dzięki temu dopiero podczas pakowania można dołączać do gotowej części programu demonstracyjnego odpowiednią muzykę czy też grafikę (oczywiście przystosowaną do wykorzystania z wcześniej przygotowanym kodem).

C-64

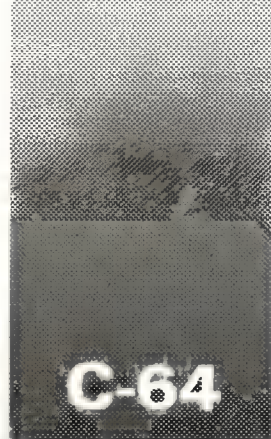
Oprócz wymienionych umieściliśmy także kilka interesujących programów demonstracyjnych. Pierwszy z nich to ostatnia produkcja dwóch ludzi: Rekina z grupy Parados oraz Carriona z grupy Elysium dla grupy Asphyxia. Demo to zostało wydane na warszawskim party, które odbyło się w marcu tego roku. Program składa się z trzech części. Głównym efektem pierwszej z nich jest dość duży scroll poruszający się także po ramce. Kolejna część to poruszanie rysunku w trybie multicolor. Ostatnia część dema jest z pewnością najciekawsza, zarówno ze względu na dobrą grafikę oraz zastosowany ciekawy efekt. Część ta to rysunek o długości dwóch ekranów wykonany w technice FLI poruszany w prawo i w lewo. Efekt ten jest dość trudny do osiągnięcia ze względu na konieczność szybkiego przepisywania dość dużej ilości danych oraz odpowiedniego dobierania czasów (tzw. cyklowanie).

Kolejne demo to również polska produkcja, tym razem grupy Charged. W stosunku do ich poprzednich prac demo to jest już dużo lepsze i ciekawsze. Znacznie poprawiła się jakość zastosowanej grafiki, a także poziom kodu i staranność wykonania. Demo to jest jedynie tzw. demem kontaktowym, czyli zawierającym adresy członków grupy pragnących nawiązać kontakt z innymi ludźmi ze sceny C64.

Ostatnim umieszczonym na lipcowym PDP programem demonstracyjnym jest mega-demo grupy Babygang pod tytułem Hexagone II. Z pewnością warto zwrócić uwagę na kilka ciekawych efektów zawartych w częściach tego dema jak na przykład ruchoma płaszczyzna składająca się z dużej ilości kropek, animowana postać karateki przeniesiona z Amigi, czy też ciekawy scroll do góry. Całość jest bardzo starannie wykonana, nie ma żadnych widocznych błedów ani niedociągnięć. Na uwagę zasługuje także dobra muzyka, po części przeniesiona z Amigi.

Mam nadzieję, że umieszczone przez nas programy użytkowe okażą się pomocne w waszej pracy, a demonstracyjne dostarczą dużo ciekawych wrażeń.

Jarosław "Jarri" Horodecki



MAPA PAMIĘCI

STRONA 1: $0100 DO $01FF

$0ff-$10a	255-266	obszar używany przy konwersji liczb zmiennoprzecinkowych na stałe łańcuchowe
$100-$13e	256-319	tabela błędów odczytu taśmy
$140-$1ff	320-511	Basic: obszar stosu pamięci

STRONA 2: $200 DO $2FF

$200-$258	512-600	bufor edytora
$259-$262	601-610	tabela do 10-ciu aktywnych numerów plików logicznych
$263-$26c	611-620	tabela urządzeń zewnętrznych przypisanych do odpowiednich plików logicznych
$26d-$276	621-630	tabela wtórnych adresów każdego z plików
$277-$280	631-640	bufor klawiatury (do 10 znaków)
$281/$282	641/642	standardowy adres początku programów w Basic'u (zwykle $800)
$283/$284	643/644	wskaźnik ostatniej - będącej do dyspozycji Basic'a - komórki RAM'u (zwykle $a000)
$285	645	flaga TIME-OUT dla portu szeregowego
$286	646	aktualny kod koloru drukowanego tekstu
$287	647	kolor znaków pod kursorem
$288	648	starszy bajt adresu pamięci ekranu
$289	649	wielkość bufora klawiatury
$28a	650	flaga samopowtarzania - dotyczy klawiszy kursora, INSERT ,DEL, SPC (128 = dla wszystkich klawiszy,64 = żadne)
$28b	651	szybkość samopowtarzania
$28c	652	opóźnienie samopowtarzania
$28d	653	flaga klawiszy SHIFT(1), C=(2), CTRL(4).
$28e	654	ostatnio używane klawisze kontrolne
$28f/$290	655/656	wektor procedury stanu klawiatury
$291	657	flaga: 0=przełączanie duże/małe litery SHIFT/C=aktywne, 128=nie aktywne
$292	658	flaga scrolowania
$293	659	RS232: wizerunek rejestru sterującego 6551
$294	660	RS232: wizerunek rejestru rozkazów 6551
$295/$296	661/662	RS232: niestandardowa wartość wskaźnika transmisji
$297	663	RS232 - rejestr statusu (ST)
$298	664	RS232: liczba na sekundę odbieranych bitów
$299/$29a	665/666	RS232: stała taktów sterujących
$29b	667	SIARS232: wskaźnik końca bufora wprowadzania
$29c	668	RS232: wskaźnik początku bufora wprowadzania
$29d	669	RS232: wskaźnik końca wyjścia bufora
$29e	670	RS232: wskaźnik początku bufora wydawania
$29f/$2a0	671/672	pamięć chwilowa wektora IRQ podczas korzystania z magnetofonu
$2a1-$2a5	673-677	pamięć chwilowa flagi CIA podczas korzystania z magnetofonu
$2a6	678	flaga wersji (1=PAL, 0=NTSC)
$2a7-$2ff	679-767	nie używane

STRONA 3: $0300 DO $03FF

$300/$301	768/769	wektor wydawania komunikatów ($e38b)
$302/$303	770/771	wektor procedur wczytywania lub wyprowadzenia linii w Basic'u
$304/$305	772/773	wektor procedur przekształcenia interpretacji kodu ($a57c)
$306/$307	774/775	wektor procedury LIST ($a71a)
$30a/$30b	778/779	wektor analizy wydruku ($ae86)
$30c	780	akumulator przy instrukcji SYS
$30d	781	rejestr X przy instrukcjach SYS
$30e	782	rejestr Y przy instrukcjach SYS
$30f	783	status procesora przy instrukcjach SYS
$310	784	rozkaz JMP dla funkcji USR ($4c)
$311/$312	785/786	wektor USR (adresy funkcji USR)
$313	787	nie używane
$314/$315	788/789	wektor IRQ ($ea31)
$316/$317	790/791	wektor BRK ($fe66)
$318/$319	792/793	wektor NMI ($fe47)
$31a/$31b	794/795	wektor OPEN ($f34a)
$31c/$31d	796/797	wektor CLOSE ($f291)
$31e/$31f	798/799	wektor CHKIN ($f20e)
$320/$321	800/801	wektor CHKOUT ($f250)
$322/$323	802/803	wektor CLRCH ($f333)
$324/$325	805/805	wektor CHRIN ($f157)
$326/$327	806/807	wektor CHROUT ($f1ca)
$328/$329	808/809	wektor STOP ($f6ed)
$32a/$32b	810/811	wektor GETIN ($f13e)
$32c/$32d	812/813	wektor CALL ($f32f)

$32e/$32f	814/815	wektor USER (użytkownik może zdefiować samemu)
$330/$331	816/817	wektor LOAD ($f4a5)
$332/$333	818/819	wektor SAVE ($f5ed)
$334-$33b	820-827	nie używane
$33c-$3fb	828-1019	bufor kasetowy
$3fc-$3ff	1020-1023	nie używane

ZAKRES PAMIĘCI OD $0400 DO $FFF

$0400-$07e7	1024-2023	normalny zakres pamięci ekranowej
$07f8-$07ff	2040-2047	wskaźnik bloków danych sprite'ów (jeśli ekran rozpoczyna się od $400)
$0800-$9fff	2048-40959	program Basic'a oraz zmienne
$a000-$bfff	40960-49151	Basic-ROM lub RAM lub tryb ROM
$c000-$cfff	49152-53247	wolny RAM dla procedur maszynowych itp.
$d000-$dfff	53248-57343	zakres wejścia/wyjścia lub generator znaków lub RAM
$e000-$fff	57344-65535	***kernal***ROM lub ROM lub tryb ROM



I/O I ZAKRES GENERATORA ZNAKÓW

$d000-$d02e	53248-53294	układ VIC
$d400-$d41c	54272-54300	układ SID
$d800-$dbe7	55296-56295	kolor-RAM
$dc00-$dc0f	56320-56335	CIA #1
$dd00-$ddof	56576-56591	CIA #2
$d000-$d7ff	53248-55295	generator znaków dużego pisma
$d800-$dfff	55296-57343	generator znaków małego pisma

opr. R.G.



C-64

DYSK cz. 2



EX-DOS

Zgodnie z obietnicą w opisie lipcowego PDP prezetujemy w tym artykule opis świetnego, moim zdaniem, programu do edycji zawartości dyskietki. Program ten jest dość ciekawy ze względu na kilka bardzo przydatnych i żadko spotykanych możliwości. Mimo, że pochodzi on już z mocno zakurzonej dyskietki to jednak w dalszym ciągu nie ma sobie równego programu. Oczywiście jak każdy program komputerowy, tak i ten nie jest pozbawiony pewnych wad, ale o tym później.

Oprócz kilku opcji takich jak inicjalizacja odpowiedniego napędu, czy też zmiana kolorów ekranu i czcionki, pierwszą opcją programu jest wczytanie zwykłego katalogu dyskietki i wyświetlanie go. Funkcja ta jest bardzo przydatna przy szukaniu dyskietki, na której będziemy chcieli przeprowadzić konkretne operacje.

Kolejna opcja w menu to wczytanie tzw. rozszerzonego katalogu dysku (extended directory). W rzeczywistości jest to właściwie całkiem niezły edytor do zmieniania zawartości katalogu. Za jego pomocą możemy całkowicie zmienić zawartość całego katalogu dyskietki. Edycja może się odbywać na dwa sposoby.

Pierwszy z nich to działania na całym katalogu równocześnie, drugi daje możliwość operowania na zadanym pliku. Niektóre z opcji działają jednak w obu przypadkach:

L - włączenie zabezpieczenia danego pliku lub wszytkich plików.
U - wyłączenie zabezpieczenia danego pliku lub wszystkich plików.
B - nadanie plikowi pełnej nazwy o długości 16 znaków (dodanie spacji).
N - skrócenie nazwy pliku o niepotrzebne spacje.
T - wyliczenie prawdziwej długości pliku lub wszystkich plików na dysku.
R - ponowne odczytanie zawartości katalogu dyskietki.

Jest też kilka opcji które działają tylko przy ogólnej edycji całego katalogu:

W - zapisanie zawartości katalogu na dyskietce.
V - "odśmiecenie" dyskietki.
S - usunięcie z katalogu wszystkich plików typu DEL.
M - wejście w tryb edycji.
^ - wyjście do głównego menu bez zgrywania katalogu na dysk.

Są też rozkazy dostępne tylko w trybie edycji plików:

Z - skasowanie wszystkich plików poniżej zaznaczonego.
INS - wstawienie jednego pustego pliku do katalogu.
DEL - skasowanie aktualnie zmienianego pliku.
^ - wyjście z trybu edycji plików.

W trybie edycji przez naciśnięcie kursora w prawo wchodzimy w tryb edycji zaznaczonego pliku. Możemy wtedy po kolei zmienić wszystkie parametry pliku takie jak jego nazwę, ilość bloków jaką chcemy aby była wyświetlana w katalgu, typ tego pliku (spośród PRG, DEL, SEQ, REL oraz USR), początek tego pliku na dysku (jako scieżka oraz sektor) oraz adres, pod jaki plik ma być załadowany do pamięci. Gdy wszystkie te dane podamy odpowiednio przerobiony plik zostanie zapisany w pamięci. Zapis tychże danych na dysk możliwy jest po użyciu opcji W z poziomu edycji całego katalogu. Wadą edycji katalogu jest brak możliwości zmiany nazwy dysku oraz jego identyfikatora.

Przejdźmy z powrotem do menu głównego. Kolejną jego opcją jest edycja BAMu, czyli obszaru dysku, w którym komputer przechowuje dane o zajętości poszczególnych sektorów. Po wejsciu do edycji BAMu ukazuje się jego mapa z gwiazdkami, które oznaczają sektory zajęte już przez dane, a kropki - sektory wolne. Oczywiście zawartość BAMu, jak przystało na edytor, możemy dowolnie modyfikować (w tryb edycji wchodzimy za pomocą klawisza M). Aby uzyskać nieco bardzie przejrzysty obraz BAMu możemy jeszcze za pomocą klawisza E wybrać opcję Extended BAM, która pokazuje także sektory, które były już zajęte przez jakieś dane, a które zostały następnie usunięte. Sektory te oznaczone są literą S.

Podczas edycji BAMu możemy używać dwóch komend

A - aby dany sektor uznać za zajęty

F - aby dany sektor był wolny do zapisu.

Po wykonaniu całkowitej edycji możemy zgrać gotowy BAM na dysk za pomocą komendy W.

Nie polecałbym jednak, aby po losowej zabawie z BAMem na dysku zawierającym istotnie dla nas dane próbować coś zapisywać, gdyż może się to skończyć dla. Mało wprawionym użytkownikom C64 nie polecam zabawy z programami w rodzaju EX-DOSa na dyskach z ważnymi plikami. Najlepiej jest przygotować sobie specjalną dyskietkę, na której będziemy ekperymentować.

Po edycji BAMu przyszedł w końcu czas na edycję poszczególnych sektorów dyskietki. Służy do tego opcja S wybierana z głównego menu programu. Po jej włączeniu możemy wybrać dowolny sektor dysku, który chcemy zmieniać. Standardowo ustawiona jest ścieżka 18 sektor 0. Dane na dysk możemy wpisywać zarówno w postaci dziesiętnej, jak i szesnastowej. Edycja danych w każdym z sektorów odbywa się za pomocą klawiszy kursora, a uaktywniana jest za pomocą klawisza M.

Z trybu edycji wychodzimy jak ze wszystkich innych opcji tego programu przez naciśnięcie ^. Korzystając z opcji T, odczytujemy wszystkie połączenie sektorów, następnie możemy korzystać z dwóch dodatkowych funkcji: F - wyświetlenie pierwszego sektora należącego do pliku oraz L - wyświetlenie ostatniego sektora należącego do pliku. Istnieje także możliwość kopiowania sektorów z dowolnego miejsca na dysku w dowolne inne miejsce (klawisz C) oraz prostego przejścia do następnego sektora na dyskietce (klawisz N). Program umożliwia także wykonywanie operacji na całych ścieżkach dyskietki. Można je odczytywać, kopiować w dowolne inne miejsce na dyskietce, a także przeformatowywać daną ścieżkę i wykonywać na niej ponowny zapis.

Ostatnie dwie opcje progamu różnią się nieco od pozostałych, gdyż służą do przeprowadzania operacji nie na dyskietce, ale na pamięci stacji dysków. Pierwsza z opcji (klawisz X) pozwala na edycję buforów stacji, natomiast druga (klawisz M) to edycja pamięci RAM procesora stacji. Możliwość ta może się okazać bardzo przydatna dla bardzie zaawansowanych użytkowników, gdyż dzięki niej można wykonywać za stacją dysków wiele ciekawych sztuczek.

Mam nadzieję, że opis tego programu okaże się wam przydatny. W następnym odcinku powrócimy do naszych zmagań z dyskietką.

Jarosław "Jarri" Horodecki

C-64

Nowa taśma



PUBLIC DOMAIN

DARKSIDE #04 - grupy PARADOS

SID'N'VIC #01

STYLE #01 - grupy SKYLIGHT

CIABACH #04/05 - grupy CRAZY BOYS

HIGHLIFE #011 - grupy ELISIUM

AXEL NEW #005 - grupy AXEL

PEZ IN MY BRAIN - demo grupy Panoramic Design

3X3 Editor - edytor czcionek grupy Skylight

The Addy Corner - magazyn z adresami

Handy Term V.8.4 - terminal

Humor Basic

Gandlaf Protector - do zabezpieczania programów

M&C Editor - do edycji komunikatów i komend z ROM'u

Last Work 4 Asx - trzyczęściowe demo grupy Asphyxia

O zasadach nabywania naszych PDP piszemy na str. 22.

C-64



Programy ciekawe, zwariowane i takie sobie

Prezentujemy dziś kolejną porcję programów nadesłanych przez naszych czytelników.

Jako pierwszy cytujemy list Grzegorza z Rumii - Janowa, który nadesłał kolejne propozycje swych programów.

/.../ Pierwszy z nich to inne rozwiązanie czyszczenia ekranu:

```
10 FOR T=40 TO 0 STEP -1: FOR X=39 TO 0 STEP 1
20 POKE 1024=T*40=X,32: POKE 1024+(5+T)*40+X,32
30 POKE 1024+(10+T)*40+X,32: POKE
   1024+(15+T)*40+X,32
40 POKE 1024+(20+T)*40+40+X,32: NEXT X: NEXT T
50 PRINT CHR$ (147);
```

Drugi program jest pozbawiony wady, z którą nie mógł sobie poradzić Grzegorz z Ropczyc (nr 1/92). Losuje on sześć liczb (w zakresie 1-49), lecz w jednej szóstce nigdy nie będzie dwóch identycznych cyfr.

```
10 PRINT CHR$(147): DIM T(6): FOR X=1 TO 6
20 T(X)=INT(RND)(0)*49): V=1
30 IFV>X-1 THEN 60
40 IF T(V)=T(X) THEN 20
50 V=V+1: GOTO 30
60 NEXT: FOR Z=1 TO 6: PRINT T(Z); ",";: NEXT
70 GET A$: IF A$="" THEN 70
80 GOTO 10
```

Trzeci program oblicza dowolny procent z liczby, a czwarty, jaki procent liczby pierwszej stanowi druga:

```
10 INPUT "PODAJ LICZBE ";A
20 INPUT " ILE PROCENT TEJ LICZBY OBLICZYC ";X
30 T=A/100: W=T*X
40 PRINT X:"% LICZBY";A;"= ";W
```

```
10 INPUT "PODAJ 1 CYFRE";A
20 INPUT "PODAJ 2 CYFRE";B
30 T=A/B: P100/T
40 PRINT B; "TO";P; "% LICZBY"; A
```

Piąty program to moja wersja zagara czasu rzeczywistego.

```
10 INPUT "GODZINA"; G%: IF G%>23 OR G%<0
   THEN 10
20 IF G%<10 THEN G% THEN G$="0"+
   RIGHT$(STR$(G%),1): GOTO 40
30 G$=RIGHT$ (STR$(G%),2)
40 INPUT "MINUTA"; IF M%>59 OR M%<0 THEN 40
50 IF M%< 10 THEN M$="0"+ RIGHT$
   (STR$(M%),1):   GOTO70
60 M$=RIGHT$ (STR$ (M%),2)
70 INPUT "SEKUNDA";S%: IF S%>59 OR S%<0
   THEN 70
80 IF S%<10 THEN S$="0"+ RIGHT$(STR$(S%),1):
   GOTO 100
90 S$=RIGHTS (STR$(S%),2)
100 TI$=G$+M$+S$: PRINT CHR$ (147);
110 PRINT CHR$(19); MID$(TI$,1,2);":"; MID$
    (TI$,3,2);":"; MID$ (TI$,5,2): GOTO 110
```

Kolejny program to mini baza danych.

A) Program pobierajcy i zapisujcy dane.

```
10 INPUT "ILE ZDAN WPISUJESZ ";A: DIM T$(A)
20 FOR T=1 TO A
30 PRINT "ZDANIE";T: INPUT T$(T)
40 OPEN 1,1,1, "DANE": PRINT #1,A
50 FOR T=1 TO A
60 PRINT #1,T$(A): NEXT
70 CLOSE 1
```

B) Program odczytujący i wyświetlający dane.

```
10 CLOSE 1: OPEN 1,1,0,"DANE"
20 INPUT #1,A
30 DIM T$(A): FOR T=1 TO A
40 INPUT #1,T$(T): NEXT
50 CLOSE 1
60 FOR T=1 TO A: PRINT T$(T);: NEXT
```

Poniższy program umożliwia zapalenie dowolnego bitu w bajcie.

```
10 INPUT "KTORA KOMORKE PAMIECI CHCESZ
   ZMODYFIKOWAC ";A: B=65535
20 IF B-A OR B-AB THEN RUN
30 PRINT "W KOMORCE ";A;" ZNAJDUJE SIE
   WARTOSC ";PEEK(A)
40 INPUT "KTORY BIT W TYM BAJCIE CHCESZ
   ZAPALIC ";X
50 IF X ORX8 THEN 40
60 Q=256; FOR T=8 TO X STEP -1: Q=Q/2
70 NEXT: POKE A,PEEK(A) OR Q
```

Następny program umieszcza w określonym obszarze pamięci tekst, co jest ogromnym ułatwieniem np. przy pisaniu scroll'i. Ostatnią daną musi być słowo "END".

```
10 INPUT "OD JAKIEGO ADRESU UMIESZCZAC
   W PAMIECI TEKST";ADR
20 READ A$: IF A$="END" THEN POKE ADR,0: END
30 FOR  T=1 TO LEN(A$)
40 POKE ADR,ASC(MID$(A$,T,1)): ADR=ADR+1
50 NEXT: GOTO 20
100 DATA "TU WPISZ SWOJ TEKST"
1000 DATA "END"
```

Kolejny program jest całkiem zwariowany. Co każde przerwanie IRQ zmienia kolor kursora.

```
10 FOR T=0 TO 18: READA: POKE 49152+T,A
20 NEXT: SYS 49152
```

```
30 DATA 120,169,13,141,20,3,169,192,141,21
40 DATA 3,88,96,238,134,276,48,234
50 CODING BY ORION/ANTRAX
```

Oto listing procedury w kodzie maszynowym:

```
COOO  SEI
      LDA#$OD
      STA$0314
      LDA#$OC
      STA$0315
      CLI
      RTS
      INC$0286
      JMP$EA31
```

Ciekawy efekt (po uruchomieniu programu) daje np. pisanie czegoś, lub wylistowanie programiku.

A teraz trzy programiki czyszczące ekran w ciekawy sposób:

Listing 1

```
10 FOR T=0 TO 19: FOR X=0 TO 24
20 POKE 1984+X840+T,32: POKE 2023+X*40-T,32
30 NEXT X,T: PRINT CHR$(147)
```

Listing 2

```
10 FOR X=0 TO 24:FOR T=0 TO 19
20 POKE 1024+X*40+T,32: POKE 1063+X*40-T,32
30 NEXT T,X: PRINT CHR$(147);
```

Listing 3

```
10 POKE 56325,0
20 PRINT CHR$(147);
30 POKE 56325,51
```

Następny program to scroll w Basic'u. Jest to przerobiona na C-64 wersja programu pt. "Pętla" z Bajtka, nr 3-4186. Ostatnią daną musi być słowo END.

```
10 DIM$(1000): PRINT CHR$(147);
20 GOSUB 120
30 P$=LEFT$(T$(T),40)
40 T$(T)=RIGHT$(T$(T),LEN(T$(T)0-40)
50 PRINT CHR$(19);P$
60 P$=RIGHT$(P$,39)=LEFT$(T$(T),1)
70 T$(T)=RIGHT$(T$(T),LEN(T$(T))-1)
80 IF LEN(T$(T))=0 THEN T=T+1
90 IF T$(T)="END" THEN GOSUB 120
100 FOR X=1 TO 100: REM PETLA OPOZNIAJACA
110 GOTO 50
120 RESTORE: T=0
130 READ T$(T): IF T$(T)="END" THEN 150
140 T=T+1: GOTO 130
150 T=0: RETURN
200 DATA "POZDROWIENIA DLA EX-CZLONKOW
    BYLEGO QUARTETU "
210 DATA "I RESZTY REDAKCJI
    C64/+4 & AMIGA"
1000 DATA "END"
```

Ostatni program nazywa się "Zbuntowany komputer", a jak działa, przekonajcie się sami.

```
10 GET A$: IF A$="" THEN 10
20 A=ASC(A$)+INT(RND(TI)*256)
30 IF A255 THEN A=ABS(255-A)
40 PRINT CHR$(A);: GOTO 10
```

Dziękujemy kol. Grzegorzowi za ten pokaźny pakiet programów.

Kol. Artur z Łodzi nadesłał progrm, o którym pisze, że:

...jest tylko zwariowany i chyba nie nadaje się do niczego. Przeznaczony jest dla posiadaczy Datasette. Proszę uruchamiać go w ciszy.

```
5 FOR B=U TO 10
10 READ A
20 FOR C=U TO 9: POKE 1,7: NEXT
30 GOSUB 100
```




```
40 NEXT: RESTORE: GOTO 5
100 FOR N=U TO A: NEXT: RETURN
110 DATA200,200,100,100,200,100,100,100,200,
    100,200
```

Kolejne programy nadesłały Joanna i Iwona z Konina. Brawo dla pierwszych, odważnych dziewczyn!

Program pierwszy testuje szesnaście kolorów obwoluty, oraz wpisuje nam nr koloru ramki zewnętrznej i po kolei numery zmieniających się kolorów obwoluty.

```
10 PRINT CHR(147): POKE 53281: POKE 646,1: PRINT
20 PRINT "** TLO: KOLOR NR= ";PEEK(53281)
   AND 15
25 PRINT: PRINT: POKE 646,0
30 FOR B=0 TO 15
35 POKE 53280,B
40 PRINT "*** OBWOLUTA: KOLOR NR= ";
   PEEK(53280) AND 15
45 FOR X=1 TO 2000: NEXT X
50 NEXT B
55 END
```

Kolejny to program malujący trzema kolorami.

```
10 PRINT CHR$(147): POKE 53280,0:
   POKE    53281,0
20 FOR A=1024 TO 2023 STEP 2
30 POKE A,160: POKE 54272+A,1: NEXT A
40 FOR B=1025 TO 2023 STEP 2
50 POKE B,160: POKE 54272+B,2: NEXT B
60 FOR C=1024 TO 2023
70 POKE C,160: POKE 54272+C,5: NEXT C
80 POKE 53280,5
90 FOR X=1 TO 1500: NEXT X 100 GOTO 10
```

Kol, Piotr z Krakowa pisze:

Po kilkakrotnym przeczytaniu rubryki "Programy ciekawe..." postanowiłem napisać własną wersję scrolla. Jego zaletą jest to, że można umieścić na ekranie tekst o dowolnej szerokości ustawianej ilością spacji w łańcuchu D$. Tekst może mieć długość do 255 znaków i może zaczynać się w dowolnym miejscu ekranu (linia nr 5). Można także zmieniać prędkość jego przesuwu (pętla w linii 6). A oto sam program:

```
1 A$="PIOTR BOCHACZYK PREZENTUJE PLYNNE
  PRZESUWANIE TEKSTU (C) 1992 - RUN-SOFT"
2 D$="          ": REM 10 SPACJI
3 FOR A=1 TO LEN(A$)
4 D$=RIGHT$(MID$(A$,A,1), LEN(D$))
5 CHR$(19) " - - I I"D$
6 FOR L=0 TO 100: NEXT: REM OPOZNIENIE
7 NEXT: GOTO 3
```

Znaki - i I oznaczają odpowiednio: kursor w dół i w lewo. Znaków tych można tam umieścić potrzebną użytkownikowi ilość.

Dziękujemy wszystkim autorom. Zapraszamy do dalszej wspólnej nauki i zabawy!

Redakcja

Czytajmy szybciej...

C-64

Prezentowany obok program nadesłał nam pan dr. Joachim Kucias. W swoim liście pisze: "Ćwiczenia tym programem umożliwiają znaczne skrócenie czasu czytania i są dodatkowo ćwiczeniem koncentracji. Sądzę, że program będzie pomocny użytkownikom C=64."

```
10 REM *** TEMPOREADER ***
20 REM **KURS SZYBKIEGO CZYTANIA/C=64**
30 REM **J. KUCIAS/1992**
40 GOTO460
50 A$="":B$=" ":C$="<control/9><control/0>:A1$="":A2$="":NZ=0
60 FORI=1TON
70 L=INT(X0+X1*RND(1)):A$=A$+CHR$(L):A1$=A1$+CHR$(L):A2$=A2$+C$:NZ=NZ+1
80 IFN=IORLK=0THEN100
90 FORJ=1TOLK:A$=A$+B$:NZ=NZ+1:NEXTJ
100 NEXTI
110 RETURN
120 :
130 LE=INT((39-LEN(A$))/2)+1:POKE781,20:SYS59903
140 POKE214,14:POKE211,LE:SYS58732
150 PRINTA$:T=TI
160 IFTI-T<ZTHEN160
170 POKE781,14:SYS59903:U=28:V=49:GOSUB360:GOSUB410
180 POKE214,20:POKE211,0:SYS58732
190 PRINT"WPISUJ PRZECZYTANE LITERY :";A2$:A0$=""
200 POKE214,20:POKE211,27:SYS58732
210 FORI=TON:POKE198,0
220 GETQ$:IFQ$=""THEN220
230 PRINTQ$:POKE214,20:POKE211,27+I:SYS58732:A0$=A0$+Q$
240 NEXTI
250 IFA1$=A0$THEN270
260 POKE53280,10:W=W+1:GOTO280
270 POKE53280,5
280 FORG=1TO150:NEXTG:POKE53280,12
290 RETURN
300 :
310 FORI=0TO19:J=39-I
320 POKE55776+I,13:POKE55776+J,13:POKE55936+I,13:POKE55936+J,13
330 POKE1504+I,160:POKE1504+J,160:POKE1664+I,160:POKE1664+J,160
340 NEXTI:RETURN
350 :
360 FORI=19TO0STEP-1:J=39-I
370 POKE55776+I,0:POKE55776+J,0:POKE55936+I,0:POKE55936+J,0
380 POKE1504+I,160:POKE1504+J,160:POKE1664+I,160:POKE1664+J,160
390 NEXTI:RETURN
400 :
410 S=54272:FORI=STOS+24:POKEI,0:NEXTI
420 POKES+1,U:POKES,V:POKES+3,12:POKES+2,128:POKES+5,8:POKES+6,28:POKES+4,65
430 FORI=15TO5STEP-0.5:POKES+24,I:NEXTI
440 POKES+4,0:RETURN
450 :
460 PRINTCHR$(147):POKE53280,12:POKE53281,12:PRINTCHR$(8)
470 POKE214,3:POKE211,19:SYS58732:PRINT"<control/1>KURS"
480 POKE214,5:POKE211,12:SYS58732:PRINT"SZYBKIEGO CZYTANIA"
490 POKE214,7:POKE211,17:SYS58732:PRINT"J. KUCIAS"
500 POKE214,9:POKE211,13:SYS58732:PRINT"POZIOM (1-5):":POKE198,0
510 GETQ$:IFQ$=""THEN510
520 P=VAL(Q$):IFP<ORP>5THEN510
530 POKE214,9:POKE211,28:SYS58732:PRINTQ$
540 K=O:N=P+1:LK=0:XO=64.9999975:X1=25.000005:W=0
550 LG=INT((39-N)/(N-1)
560 Z=INT(12*N+NZ/3):Z3=INT(3*N+NZ/6)
570 WZ=W
580 FORM=1TO2:U=42:V=62
590 GOSUB50:GOSUB310:GOSUB410:GOSUB130
600 K=K+1:IFK>999THEN690
610 NEXTM:POKE781,23:SYS59903
620 AK=INT(W/K*100):POKE214,23:POKE211,17:SYS58732
630 PRINTAK;" [illegible] BLEDOW"
640 IFZ<Z3THEN660
650 Z=Z-2:GOTO570
660 IFLK>=LGTHEN690
670 LK=LK+1:GOTO560
680 :
690 PRINTCHR$(147):POKE214,20:POKE211,6:SYS58732:IFAK>10THEN710
700 PRINT"PRZEJDZ DO POZIOMU ";N;" !":END
710 PRINT"POWTORZ POZIOM ";P;" !":END
720 :
```

PUBLIC DOMAIN PACK

PUBLIC DOMAIN PACK C-64

Styczeń '91 (nr 1)

- Mega demo grupy „VISION"-MIST2, Preview do gier: UN SQUADRON, PUZZLENOID, TURRICAN.

Luty '91 (nr 2)

- Tune of month, Logo Writer V 2.0, Fast Cruelcrunch, WRATH+ (DEMO)[02], Dreptacz - BASIC, SWISS CHEESE/CFA, Disk Fast Loader.

Marzec '91 (nr 3)

- Font Grub 1.0, Projektant Duszków, Strzałka 64+, Piratek - gra, V4.0 - Symphonies, Cruiser, The First, Commercial Break, Relakator 64, Korektor 64, Flash, HOT SHOT nr 9 (zach. mag.), BAD NEWS nr 2 - j.w., demo - rekord - 290 sprite'ów!, demo: NEW INTRO, demo: LET'S DYCP, Kontakt Corner - adresy, New Fast - działa z 1541 i 1541 II, CSLINKER V2.0.

Kwiecień '91 (nr 4)

- Digi - Organizer - program do tworzenia muzyki z użyciem digitalizacji dźwięku, „ONE YEAR - RADIUS" - mega demo grupy RADIUS.

Maj '91 (nr 5)

- CRUEL SOLIDERS - demo, DESTINATION - demo, SUCKER DJ! - demo (digi mix), MUSIC SEARCHER - do wycinania ilustracji muzycznych z programów, MEGA DEMO „INFOSYSTEM 91".

Czerwiec '91 (nr 6)

- Fonteditor, Sindata Editor, Color Editor, Disk - Noter, Gwiazdy - demo graficzne, FILGRAEPH 2.2/BML, NOTE TO FLI V 2.2, AFLI - EDITOR V 1.2, RESET - MON,8,1, TURBO - ASS 5, HIGHLIFE #5, AXEL NEWS #1, DISK NOTKA/PADUA, PSC - MAG #9'06/91, CONSPIRE? OREGON - demo, CONTACT DEMO/ORE, SHOWPIX.

Lipiec '91 (nr 7)

- Mega demo „MY, OH MY!" grupy LIGHT, Game Music Composer - edytor muzyczny grupy GRAFFITY z Węgier.

Sierpień '91 (nr 8)

- MegaDemo „Unnamed" grupy CAMELOT, Sound Killer - edytor muzyczny grupy TOPAZ, AFLI - edytor graficzny techniki A-FLI, Disk-Dos obsługa komend stacji dysków, Noter v2.2 grupy TOPAZ, IFFL - Squeezer kompresor dyskowy, Dismaster+ - edytor do dyskietek, Super Copy - DOS szybki program kopiujący do zbiorów, Mega Demo fińskiej grupy TOPAZ - „Graveyard Blues".

Wrzesień '91 (nr 9)

- Mega Demo grupy FLASH, Hot Shot - magazyn dyskowy, Code Sucker monitor - pr. użytkowy grupy PADUA, Mountain Ride - gra w BASIC.

Październik '91 (nr 10)

- MEGA DEMO „AIRDANCE 4" grupy T.A.T.

Listopad '91 (nr 11)

- NEW LAW & ORDER, FLT/LEGOLAND, FLT/LEGONOTE, TERMINAT .2%/FLT, SM. CRIMINAL #8, SMALL BUT FINE, HOLLY SMOKE/M12, UNITE!/SYLVIO.

Grudzień '91 (nr12)

- Armageddon 3, NOTE TO DEMO, OUTRUN 2 MUS 5 SFX, AFTER-BURNER/MON, TRIVIA-GAME MUSIC, FORM.I. SIMULATOR, 2400AD END-TUNE, NIGHTHUNTER MUSIC, TOMCAT MUS./MON, ZAMZARA TUNE/MON, NOTE TO DISK, HIGHLIFE #9.

Styczeń '92 (nr 13)

- Char Zoomer v3.1, Colour Bar Editor v3.0, Hires+A-FLI Editor, FLI Designer v1.1, Accesinus, Music Routine Cruncher v1.5, Gandalf Protector v3.0, Gandalf Coder v1.0, Disk-tape copy v.20, Gnd - packer v1.0.

Luty '92 (nr 14)

- LYNX XVI+, Sideborder Logo Editor V1.0, Intelpaint, Flieditor V3.2, 4*4 Charmaker, F(R)ONT EDITOR 3, THE GRAFIX PACK II, DEMA.

Marzec '92 (nr 15)

- Contact Dealer v3.0, Beeftrucker v1, Cross linker v3, PowerCruncer v7.1, Sample mon v.2.0, Handy Term v8.4, Highlife, Darkside, Humor Basic, dema.

Kwiecień '92 (nr 16)

- CIABACH 4&5, DARKSIDE, Convert Studio, edytor znaków grupy Skylight, Sprite Designer, Multicolour Conventer v1.3, Logo Flipper, Charset Maker v1.0.

PUBLIC DOMAIN PACK AMIGA

Styczeń '91 (nr 1)

- Programy kompresorów danych, Grafiki Borysa Vallejo, prezentacja najlepszych muzyczek, INTUITRACKER.

Luty '91 (nr 2)

- Request player; Multi ripper, 3-rd day; Phantasmagoria - demo, Master Seka; Virus Ekspert v1.6, AMOS-programy; Moduly: Killing game show, Upon Me, Let's swing it.

Marzec '91 (nr 3)

- PROTRACKER V1.0 (pakiet programowy), Najlepsze muzyczki: NOW WAIT? - DR.AWESOME, AMOS - procedury, DEMO grupy REBELES „TOTAL TRIPLE TROUBLE".

Kwiecień '91 (nr 4)

- RUBBER VECTORS - demo, KEF-TALES - demo, DISK MASTER V3.0, Moduły muzyczne: TECHNOSTYLE 2, GALAXY 2, GRAFIKA - prezentujemy rysunki - RICK PARKS.

Maj '91 (nr 5)

- VIRUS X 5.0, VIRUS TERMINATOR, PARADOX - demo, STORMCHILD - demo, Moduły muzyczne: MIAMI VOICE, ANTI ATARI SONG.

Czerwiec '91 (nr 6)

- POWER BOOT - własne menu dysku, DISK CODING SYSTEM - program do zabezpieczania dysków, Konwerter IFF - ANSI, AUER NATION - demo, Moduły muzyczne, DOCS - opis gry ELWIRA, LAMER DEFENCE - do wykrywania i niszczenia wirusów, REWENG GO OF THE LAMER - grafika w trybie D-HAM.

Lipiec '91 (nr 7)

- Sanity - demo, Amiga - Tanx (1Mb) - gra, Little Beau (1MB) - gra, There is A Light/Tonid - modules.

Sierpień '91 (nr 8)

- Real 3D - demo programu do raytraceing'u, moduł muzyczny XTC STEREO.

Wrzesień '91 (nr 9)

- Moduły muzyczne dla programu TFMX: R - TYPE; The House of Techno;VIRUS EXPERT v181 + 143; Boot Block'i, Bootx v 3.80, Imploder v 4.0.

Październik '91 (nr 10)

- ANARCHY - „THE INSPIRATION IS NONE"; ‚DUAL CREW - „NEW DIMENSION"; SANITY - „ELYSIUM".

Listopad '91 (nr 11)

- COMPUTER HEAD - animacja; CONFUSED - moduły pod medplayer i wiele sampli; ROCKED -; SAVE GAME „MONKEY ISLAND".

Grudzień '91 (nr 12)

- GEM X; BOOTX V4.13; FINAL KIT - monitor; MEGA-MON; VARIA.

Styczeń '92 (nr13)

- Super Duper 2.01; Sanity Copy; Noise Packer 3.00; Ham Sharp; Mostra; dema graf. i muz.

Luty '92 (nr 14)

- Nuke Saddam 1.4; THIEF RIPPER 2.0; DISK MASTER 3.05; ZIG ZAG #3 (grafika); dema graf. i muz.

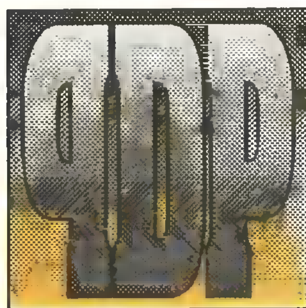
Marzec '92 (nr 15)

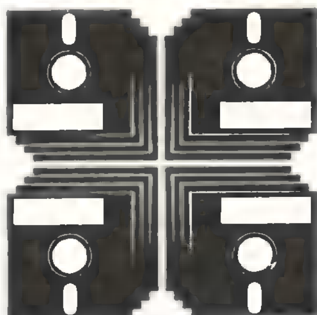
- Gry: KIM, Power Wars; PrtDrvGen; FileMaster v1.1; FixDisk!; Facc; Intro Well!; digitalizacje (IFF).

Kwiecień '92 (nr 16)

- Disk Cruncher; IBM-ST-AMIGA; HamLab v0.91; BAD v4.13; BOOTX v4.47,; DragonTiles - gra; grafiki IFF.

PUBLIC DOMAIN PACK





PUBLIC DOMAIN PACK C-64 TAPE NR 1

- TURBO
- SINUSDATA - EDITOR
- FAST CRUNCHER V3
- ANAL S.C. IBEYOND
- VECTOR - VICTORY
- PUZZLENOID+4
- TUNE OF MONTH #1
- NIM
- STRZAŁKA 64+
- LOGO - WRITER V.2.0
- CAN'T TOUCH IKUI
- NTRO PRV
- BONZIEED!!
- ZAX PACKIS
- READ THIS FIRST
- COMMERCIAL BREAK
- 290 SPRITES!
- NOTE - ABOUT
- BAD NEWS NR2
- TO BAD NEWS...
- CONTACT CORNER!
- PROJEKT DUSZKÓW
- SYMPHONY NR14
- SYMPHONY NR15
- SYMPHONY NR16
- SYMPHONY NR17
- SYMPHONY NR18
- SYMPHONY NR19
- CRUISER/GIANTS
- NOTE>ANO<PADUA
- LET'S DYSP!
- FINALTAPE
- MUSIC - SEARCHER

PUBLIC DOMAIN PACK C-64 TAPE NR 2

- TURBO
- PUBL. DOMAIN. INFO
- FONTGRUB 1.0
- DREPTACZ BASIC
- LOAD DIS FIRSY
- MACROASSEMBLER
- TURBOASSEMBLER
- RELOCATOR
- LOGOPAINTER 3!
- REASSEBLER
- SPRITE - EDITOR
- FAST - CRUEL U.2.5
- HIGHLIFE NR5
- AXEL NEWS NR1
- GWIAZDY
- FLIGRAPH 2.2/BML
- NOTE TO FLI V.2.2
- DISKNOTKA/PADUA
- MEGA PACKER/T
- MIST II/ VISION
- TTECHSCR &DYSP
- PLASMA - WORLD
- VECTORBOBS...
- VECTOR - PLOTS
- FLI - UPSCROLL
- BORDER - HIRES
- ROCK AROUND
- FACEWRITER
- CHAR EDIT 2+2
- DISKNOTER
- DESTINATION'91
- CONTACTDEMO/ORE
- FONTEDITOR
- THE END

PUBLIC DOMAIN PACK C-64 TAPE NR 3

- TURBO
- PUBLIC DOMAIN NOTE
- GRAVEYARD NOTES!
- NOTE FROM BEAT!!
- ANONYM SPEAKING!
- SNDK. V3.7/TOPAZ
- AFLI - EDITOR
- NOTER V2.2/TOPAZ
- DLW V1.5/TOPAZ
- CODE - S.MON/PADUA
- OPINION - POLL/PDA
- MOUNTAIN RAID
- PART 1
- PART 2
- PART 3
- PART 4
- PART 5
- FAIRLIGHT 1
- FAIRLIGHT 2
- FAIRLIGHT 3
- FAIRLIGHT 4
- FAIRLIGHT 5
- THE END

PUBLIC DOMAIN PACK C-64 TAPE NR 4

- TURBO
- OUT RUN 2 MUS & SFX
- AFTER BURNER/MON
- FORM.1.SIMULATOR
- 2400 AD.END - TUNE
- NIGHT HUNTER D!GI
- ELEMINATOR MUSIC
- TOMCAT MUSIX/MON
- ZAMZARA TUNE
- DYNAMIX TUNE
- HIGHLIFE NR9
- SNAKES C3
- SNARK C3
- SNERD C3
- WAREHOUSE C3
- STARTREK C3
- TOWER
- SNOOPY
- NEW LAW & ORDER
- FLT/LEGONOTE...
- TERMINAT.2%/FLT
- UNITE!/SYLVIO
- BALL - SCOPE/451
- TRIVIA - GAME MUS.
- RESET - MONITOR
- HOLY SMOKE

Zestawy „64 plus 4 PUBLIC DOMAIN PACK" można zamawiać wpłacając na konto: Bank PKO SA Oddział w Bydgoszczy konto nr: 5.09011-400522.7-2511-30-111.0 następujące kwoty: 20.000zł za pojedyńczy zestaw dyskowy dla C-64, 30.000 zł za zestaw programów PD na kasecie, 25.000zł za zestaw dla Amigi.

Blankiety wpłat powinny być CZYTELNIE wypełnione i zawierać: **imię i nazwisko, dokładny adres zamawiającego, skrót „PDP-64D"** - jeśli zamawiamy zestaw dla C-64 na dyskietce lub **„PDP-64T"** - dla zestawu taśmowego, zestaw dla Amigi prosimy zaznaczać skrótem **„PDP-A"** - dane te prosimy umieszczać na **wszystkich** odcinkach dowodu wpłaty.

W prenumeracie zestawy kosztują: **PDP-64 - 18.000zł** (12 numerów 216 tys. zl), **PDP-A - 22.000 zł** (12 numerów 264 tys. zl). Prenumeratę można zawrzeć w dowolnym terminie na okres od 3 do 12 miesięcy (do końca roku kalendarzowego). Powyższe warunki odnoszą się również do naszych zestawów wydanych w 1991r .

Zestawy taśmowe PDP-64 w 1992r. będą ukazywały się w miarę napływu nowych, ciekawych programów - o czym będziemy informować na łamach naszego pisma.

Zamów nie zwlekaj!

VOICETRACKER V4.0

C-64

Rewelacyjny program muzyczny!



Tylko 50.000 zł kosztuje fantastyczny edytor muzyczny wykorzystujący ogromne możliwości dźwiękowe komputera Commodore - 64. Oferowany zestaw zawiera dyskietkę lub taśmę magnetofonową z programem VOICETRACKER V4.0, instrukcję obsługi, oraz - dodatkowo - przykładowe demonstracje muzyczne.
UWAGA! Wersja magnetofonowa tylko 40.000 zł!

Przedsiębiorstwo ABUK posiada wyłączność na dystrybucję tego programu. Wszelkie kopiowanie programu i powielanie instrukcji jest zabronione. Nabywcy otrzymują rejestrowane kopie programu wraz z prawem nabywania nowych wersji po znacznie obniżonych cenach oraz wymiany dyskietki w razie uszkodzenia. Studiom komputerowym proponujemy zakup hurtowy (przy zakupie powyżej 10 kompletów udzielamy 20% rabatu).
Chcąc stać się posiadaczem programu VOICETRACKER V4.0 wystarczy dokonać wpłaty 50.000 zł (wersja dyskowa) lub 40.000 zł (taśma) na konto: Bank PKO SA Bydgoszcz, konto nr: 5.09011-400522.7-2511-30-111.0. Na blankiecie prosimy czytelnie podać swoje imię, nazwisko i adres wraz z dopiskiem „VV4.0" uzupełnionym literką „T" - taśma lub „D" - dyskietka.

W związku z pojawiającymi się kłopotami w dystrybucji oferowanych przez nas dyskietek i taśm (wynikającymi z nieczytelnego bądź niekompletnego wypełnienia blankietów wpłat) przedstawiamy obok specjalny druk. Blankiet ten może służyć jako zamówienie i dowód wpłaty dla wszystkich oferowanych przez nas usług: sprzdaż dyskietek i taśm PDP, Voicetracker'a, zamówienie ogłoszeń itd.

REDAKCJA

Odcinek dla wpłacającego
Zł ....
słownie ....
wpłacający ....
....
(dokładny i CZYTELNY adres)
na rachunek:
Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-2511-30-111.0.
Opłata
zł..........

Odcinek dla posiadacza rachunku
Zł ....
słownie ....
wpłacający ....
....
(dokładny i CZYTELNY adres)
na rachunek:
Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-2511-30-111.0.
Opłata
zł..........

Odcinek dla Poczty
Zł ....
słownie ....
wpłacający ....
....
(dokładny i CZYTELNY adres)
na rachunek:
Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-2511-30-111.0
Opłata
zł..........



TREŚĆ ZAMÓWIENIA:

TREŚĆ ZAMÓWIENIA:

Prosimy o CZYTELNE wypełnienie.

TREŚĆ ZAMÓWIENIA:

Prosimy o CZYTELNE wypełnienie.

Zapraszamy wszystkich do udziału w stałym konkursie pod hasłem:

Najlepszy program miesiąca

W konkursie udział mogą brać wszyscy, którzy nadeślą własne, nigdzie nie publikowane prace. Tematyka programów dowolna. Konkurs rozgrywany jest osobno dla komputerów C-64 i Amiga. Teksty programów należy nadsyłać na adres redakcji na dyskietce lub w postaci czytelnego rękopisu (dyskietki będą przez redakcję zwracane). Objętość programu wraz z opisem i komentarzem nie powinna przekraczać 4 stron maszynopisu. Raz w miesiącu Sąd Konkursowy wybierze najlepsze programy przyznając ich autorom dwie główne nagrody po

500.000 zł każda.

Decyzje Sądu Konkursowego są nieodwołalne. Oprócz zdobycia głównej nagrody autorzy mają szansę na publikację swych prac na łamach naszego pisma. Pracę prosimy podpisywać imieniem i nazwiskiem oraz dokładnym adresem autora.

Redakcja

WSZYSTKICH ZAINTERESOWANYCH NABYCIEM ZALEGŁYCH NUMERÓW

„64 plus 4 & AMIGA"

INFORMUJEMY, ŻE POSIADAMY JESZCZE OGRANICZONĄ ILOŚĆ NUMERÓW OD LISTOPADA 1990R. DO GRUDNIA 1991R.

ZAMÓWIENIA PROSIMY KIEROWAĆ NA ADRES :
Przedsiębiorstwo ABUK sp. z o.o.,
87-200 Wąbrzeźno,
ul. 1 Maja 33.
(Pod tym adresem mieści się dział kolportażu - tam też prosimy przesyłać wszelką korespondencję dotycząca kolportażu czasopisma, dyskietek, taśm itd. Adres redakcji się nie zmienił - patrz stopka.)

Wszystko o dyskietce



AMIGA

W codziennej pracy z komputerem, dyskietki odgrywają bardzo ważną rolę. Co to są dyskietki, jak funkcjonują i na co należy uważać w trakcie ich eksploatacji?

Dyskietki (zwane również dyskami elastycznymi) pojawiły się w 1973 roku. Spotyka się je wszędzie tam, gdzie istnieje potrzeba zapamiętania dużych ilości danych. Zależnie od wielkości i wykonania na jednej dyskietce można zapamiętać około 400 stron maszynopisu.

Dyskietki produkowane są z foli magnetycznej w formie krążków o średnicach:

-20 cm (Maxi - dyskietka: 8 cali)
-13 cm (Mini - dyskietka: 5 1/4 cala)
-9 cm (Mikro - dyskietka: 3 1/2 cala)

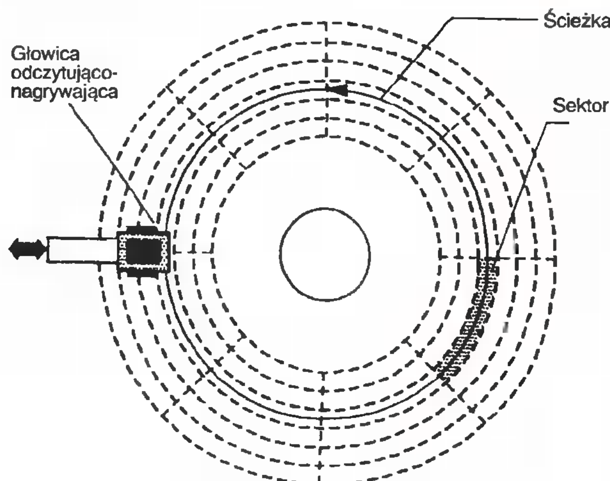
Osłona dyskietki 3 1/2 cala wykonana jest z tworzywa sztucznego. Na przedniej stronie dyskietki znajduje się miejsce dla nalepki. Zamek okienka odczytu zrobiony jest z metalu i otwiera się z chwilą włożenia dyskietki do stacji dysków, umożliwiając dostęp głowicy do nośnika.

Centralny krążek metalowy zawiera otwór centrujący i napędowy. Zabezpieczenie zapisu wykonane jest w postaci suwaka z tworzywa sztucznego.

JAK FUNKCJONUJE DYSKIETKA.

Cała zasada polega na zapisie magnetycznym. Głowica zapisująca i odczytująca jest cewką, przez którą - w czasie zapisu - płynie prąd wytwarzając pole magnetyczne. Kierunek prądu decyduje o położeniu biegunów N,S (patrz rys. 1). W trakcie odczytu dyskietki następuje sytuacja odwrotna, tzn. zmiany pola magnetycznego z dyskietki powodują indukcję prądu w cewce.

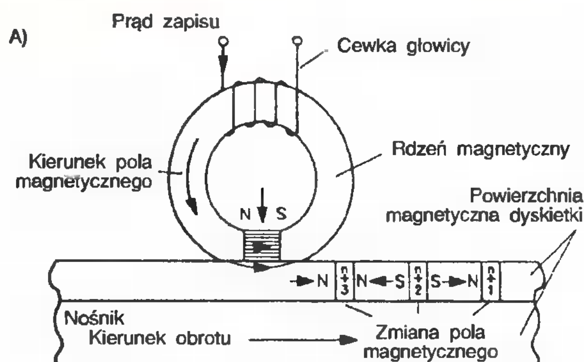
Głowica zapisuje dane na dyskietce według ściśle określonego schematu. Górna część dysku elastycznego podzielona jest na ścieżki, a te z kolei na sektory. Dyskietka Amigi ma dwie strony po 80 ścieżek (0 do 79), każda po 11 sektorów (0 do 10). W sumie daje to 1760 bloków (ponumerowanych od 0 do 1759). Pojemność dyskietki można obliczyć w następujący sposób: ilość cylindrów * ilość głowic * ilość sektorów na ścieżce * ilość bajtów na sektor (w każdym sektorze można zapamiętać 512 bajtów danych). Otrzymamy wtedy następujący rachunek: 80*2*11=1760*512=901120 bajtów. Odpowiada to 880 kB - każdy po 1024 bajty.

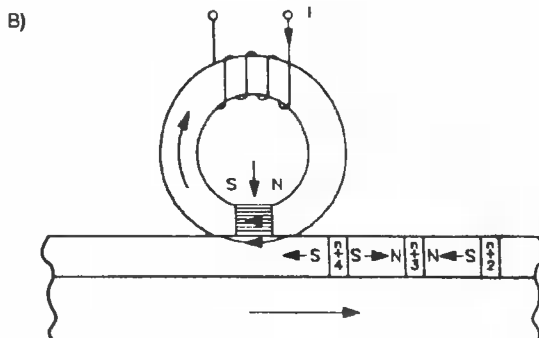


Rys. 2. Poddział dysku na ścieżki i sektory.

Na dyskietce jaden bit zajmuje 0.004 mm ścieżki. Nazwa "DyskietkaAmigi" zajmuje 14 bajtów, każdy po 8 bitów, czyli długość zapisu na ścieżce wynosi 0,05 mm. Miarą zapisu danych na dyskietce jest "BPI" (Bits per Inch - bit/cal). Współczesne napędy dyskowe pracują z 3000 do 6000 BPI.

Kto nie zna problemów przy kupnie dyskietek? Jakie dyskietki mamy kupować?





Rys. 1.

AMIGA

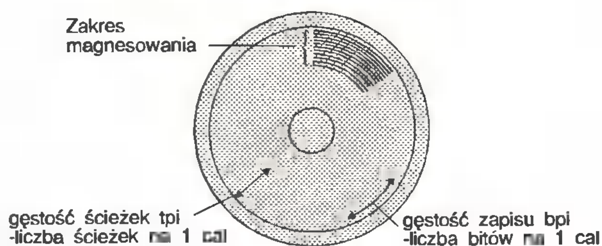
Przy nabywaniu dyskietek, poza kryterium wielkości powinniśmy również zwracać uwagę na następujące parametry:

- dyskietka jednostronna czy dwustronna?

Napędy dwustronne wyposażone w 2 głowice zapisują dane na obydwu stronach dyskietki. Dlatego zwracajmy uwagę na jej oznakowanie: 1-dla jednostronnych, 2-dla dwustronnych napędów. Podczas produkcji dyskietek są one sprawdzane przez producenta. Testowane są obydwie strony dyskietek, ale jeśli jedna z nich wykazuje wady, to dyskietka sprzedawana jest jako jednostronna. W przypadku dyskietek dwustronnych wady nie mogą znajdować się na żadnej stronie.

- gęstość ścieżek

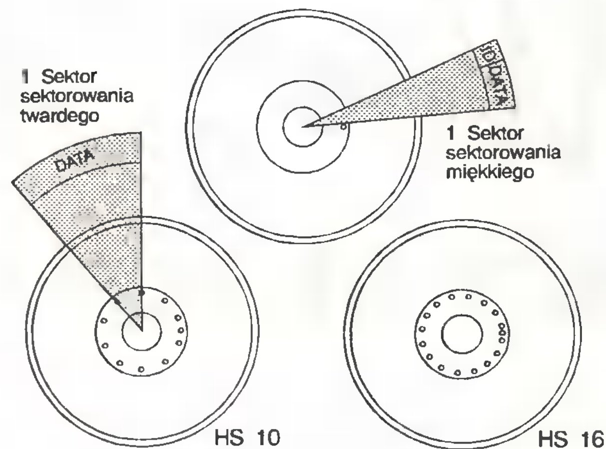
Jako gęstość zapisu oznacza się sprawdzoną gęstość w calach w kierunku promienia. Zwykle dyskietki oznaczone są jako 48, 96 lub 135 tpi (tpi=tracks per inch - ścieżka na cal). Dyskietki o większej gęstości można używać do normalnych napędów, nigdy na odwrót. Dyskietkę 96 tpi można używać na napędzie 48 tpi, natomiast nie gwarantuje się prawidłowego zapisu dyskietki 48 tpi przy próbie sformatowania jej na 96 tpi.



Rys. 3. Gęstość ścieżek (tpi), gęstość zapisu (bpi).

- formatowanie

Aby dane mogły być systematycznie zapisywane na dyskietce, górna jej część dzieli się na ścieżki a te z kolei na 10, 16 lub 26 sektorów. Ten podział (dokonywany w czasie formatowania) jest uzależniony od systemu operacyjnego komputera. Przy miękkim sektorowaniu, sektory wytworzone są przez zapis magnetyczny. Przy sektorowaniu twardym, istnieją w pobliżu otworu wewnętrznego dyskietki małe otwory sektorowe. Początek sektora rozpoznawany jest wtedy przez komórkę fotoelektryczną. Wszystkie dyskietki 3,5" calowe pracują wyłącznie z magnetycznym rozpoznawaniem sektora oraz ścieżki. 5,25" calowe dyskietki są najczęściej miękko sektorowane i mają 16 sektorów po 256 bajtów (rysunek 4). Przy miękkim formatowaniu w krążku znajduje się otwór indeksacyjny. Cały dysk zostanie podzielony na sektory podczas formatowania. Każdy sektor posiada pole identyfikacyjne. Dyskietki sektorowane twardo posiadają oprócz otworu indeksacyjnego również otwory sektorowe, a więc 10 lub 16 dla dyskietek 5 1/4". Nie występuje wtedy magnetyczne rozpoznawanie sektorów.



Rys. 4. Formatowanie twarde i miękkie.

Kto pracuje z dyskietkami zawsze ma obawy przed utratą danych. Dlatego powinno się wykonywać kopie ważniejszych dyskietek. Na trwałość dyskietek wpływ mają następujące czynniki:

- czynnik mechaniczny

Jeśli zagniemy dyskietkę to po włożeniu do napędu może wystąpić takie tarcie, że zostanie uszkodzona krawędź otworu wewnętrznego dyskietki. Dyskietka nie będzie wówczas centryczna i dane nie będą mogły być odszukane. Z dyskietkami należy obchodzić się starannie! Przy czystej głowicy w napędzie, dyskietka wytrzymuje wiele milionów przebiegów jednej ścieżki, a że dyskietka obraca się z szybkością 300 obrotów na minutę, to oznacza bezawaryjny odczyt - non stop - przez kilka miesięcy. Nigdy jednak nie używamy ciągle tej samej ścieżki przez kilka miesięcy, 24 godziny na dobę... Na przykład BASF gwarantuje średni czas pracy 70 milionów przebiegów ścieżki - to jest około 4000 godzin pracy każdej ścieżki. Z doświadczenia wynika, że ta ilość nie zostanie osiągnięta nawet po 20 latach intensywnego użytkowania! Jednakże głowica musi być bez zarzutu: nie uszkodzona i nie zabrudzona!

- przechowywanie

Dyskietki powinny być przechowywane w temperaturze od 4 do 50°C. Wilgotność powietrza powinna odpowiadać wilgotności pomieszczeń mieszkalnych. Transportowane mogą być w temperaturze od -40 do 70°C. Nie mogą być natomiast ściskane, gdyż obudowa dyskietki może się zdeformować.

- dyskietki czyszczące

Użytkownik powinien co jakiś czas oczyścić głowicę w napędzie dysków. Do tego nie trzeba wcale otwierać obudowy, gdyż istnieją specjalne dyskietki czyszczące. Następstwem zabrudzonej głowicy są występujące błędy przy nagrywaniu i odczycie.

- zakłócenia

Przez to, że dane na dyskietce zapisywane są polem magnetycznym, również pole może je skasować. Dlatego należy uważać na wszystkie źródła pola magnetycznego, które mogą się znaleźć w pobliżu dyskietki, np.: magnesy stałe, elektromagnesy (np. wkładka słuchawki telefonicznej, głośnik, itp.). Należy zachować odstęp od nich co najmniej 50 cm. Trwałe uszkodzenia mogą również wystąpić poprzez opisywanie dyskietki długopisem. Naklejki na dyskietki powinny być najpierw opisane, a dopiero potem naklejane.

Z tą podstawową wiedzą więcej już nie powinniście mieć problemów z dyskietkami.

Pamiętajcie: troska o dyskietkę to troska o zapisane na niej programy!

Opr. na podst. Amiga, nr 4/[illegible]

KĄCIK POCZĄTKUJĄCEGO KODERA cz. 15
Gadgets

AMIGA

Dzisiaj zajmiemy się pierwszą z metod komunikacji użytkownika z programem. Wszelkie programy pisane z wykorzystaniem systemu powinny być kontrolowane w jak największej części przez myszkę i tutaj bardzo ważną rolę odgrywają gadżety.

Gadżet jest to wyznaczony fragment w obrębie okna (coś w rodzaju przycisku), dla którego mamy przyporządkowaną odpowiednią funkcję, która może być wywołana poprzez naciśnięcie przycisku myszki w momence gdy kursor znajduje się w obrębie tego pola. Gadżetem jest także mały suwak, który może być przesuwany przy użyciu myszki lub pole, do którego użytkownik może wprowadzić tekst lub wartość.

Gadżety dzielą się na

- sytemowe (o których pisaliśmy już w Kąciku Początkującego Kodera)

- użytkownika, którym poświęcony jest dzisiejszy.

Gadżety użytkownika dzielą się na:

* Przełączane - gadżet, który podaje informacje o tym czy jest włączony czy wyłączony.

* Ciąg - gadżet, do którego możemy wpisać ciąg znaków.

* Wartość - działane podobne do gadżetu typu Ciąg jednak użytkownik może wpisywać wyłącznie wartości liczbowe całkowite.

* Proporcjonalne - gadżety, które dają nam do dyspozycji mały suwak, który możemy przesuwać za pomocą myszki w obrębie wyznaczonego pola.

Każdy gadżet użytkownika może zawierać grafikę oraz specjalnie stylizowaną ramkę. Możemy na przykład dla poszczególnych gadżetów przypisać obrazki o różnym znaczeniu, które mogą zostać zastąpione innymi w momencie gdy dany gadżet zostanie wybrany.

Pozycja gadżetów w oknie jest określana określona jedynie przez wymiary okna. Możemy również umieszczać gadżet względem prawej i dolnej ramki ograniczającej okno. Teraz zapoznamy się ze strukturą Gadget.

Struktura Gadget00 APTR gg_NextGadget ; wskaźnik do następnego gadżetu w liście gadżetów. Gadżety wpisujemy właśnie w taką listę to znaczy, iż poprzedni gadżet zawiera w tym polu adres następnego a ostatni zawiera w tym polu zero adres pierwszego gadżet wpisujemy w polu FirstGadget w strukturze NewWindow przefd inicjacją danego okna.

04 WORD gg_LeftEdge ; przesunięcie względem lewej krawędzi okna

06 WORD gg_TopEdge; przesunięcie względem górnej krawędzi okna

08 WORD gg_Width; szerokość gadżetu0a WORD gg_-Height; wysokość gadżetu

0c WORD gg_Flags; znaczniki określające wygląd gadżetu

0e WORD gg_Activation; znaczniki określające sposób wywoływania gadżetu

10 WORD gg_GadgetType; znaczniki określające typ gadżetu

12 APTR gg_GadgetRender; wskaźnik do struktury Boder lub Image. W zależności od tego jakie znaczniki ustawimy w polu Flags możemy otrzymać gadżet z grafiką bądź ramką lub jeżeli to pole jest ustawione na wartość równą zero to zostanie utworzony gadżet bez elementów graficznych.

16 APTR gg_SelectRender; najczęściej nie używa się tego pola gdyż określa ono kształt gadżetu po wybraniu. Częsciej używa się opcji "highlighted" zmieniającej kolory w gadżecie po jego wybraniu. Natomiast gdy chcemy aby zmieniła się grafika bądź ramka wybranego gadżetu to do tego pola wpisujemy adres struktury Border lub Image.

1a APTR gg_GadgetText; wskaźnik struktury IntuiText zawierającej tekst dla naszego gadżetu.

1e LONG gg_MutualExclude; poprzez użycie MutualExclude możemy wymusić na innych gadżetach automatyczną wyłączenie się w momencie gdy nastąpi wybranie tego gadżetu. Pole to zawiera numery gadżetów z listy gadżetów które zostaną wyłączone i tak na przykład bit numer 3 odpowiada za trzeci gadżet w liście i gdy np. dokonamy selekcji tego gadżetu a mamy ustawiony bit numer 3 w tym polu to jeżeli gadżet numer trzy był wybrany to nastąpi jego automatyczne wyłączenie.

22 APTR gg_SpecialInfo ; w tym polu umieszczamy wskaźnik do specjalnej struktury wymaganej przez gadżety typu: Proporcjonalny, Ciąg, Wartość. Jeżeli gadżet jest typu Przełączany to ta struktura nie jest konieczna (jakkolwiek można użyć struktury BoolInfo do określenia kształtu gadżetu). W przypadku gadżetu typu Ciąg lub Wartość wpisujemy tutaj adres struktury StringInfo a dla gadżetu typu Proporcjonalny wpisujemy wskaźnik do struktury PropInfo.

26 APTR gg_GadgetID; pole zawierające identyfikator gdżetu definiowany przez użytkownika

28 APTR gg_UserData ; wskaźnik lub wartość definiowana przez użytkownika. Całkowicie ignorowana przez bibliotekę intuition.

Wróćmy teraz do położenia gadżetu w obrębie okna. Jeżeli chcemy gadżet umieścić względem lewego górnego rogu okna to po prostu wpisujemy wartości odpowiadające tym przesunięciom do komórek LeftEdge i TopEdge w strukturze Gadget. Natomiast gdy chcemy umieścić gadżet w pewnej odległości od prawego dolnego rogu okna tak aby przesuwał się wraz ze zmianą wielkości okna to wpisujemy te wartości do komórek LeftEdge i TopEdge (są to wartości ujemne) oraz ustawiamy następujące znaczniki w polu Flags w strukturze Gadget:

GRELRIGHT (Gadget RELative Right) - pozycja gadżety względem prawej ramki,

GRELBOTTOM (Gadget RELative BOTTOM) - pozycja gadżetu względem dolnej ramki.

Z wymiarami gadżetu mamy podobną sytuację jak z wymiarami okna i tak na przykład jeżeli chcemy mieć gadżet o określonej wielkości to jego wymiary wpisujemy do komórek Width i Height w strukturze Gadget, natomiast jeżeli chcemy aby gadżet zmieniał swoją wielkość wraz ze zmianą okna, które zawiera ten gadżet to musimy do komórek Width i Height wpisać wartości odpowiadające szerokości i wysokości gadżetu względem wymiarów okna (wartości ujemne oznaczające o ile wymiary gadżetu są mniejsze od wymiarów okna) oraz ustawić znaczniki w polu Flags w strukturze Gadget:

GRELHEIGHT (Gadget RELative Height) - wysokość gadżetu względem wysokości okna

GRELWIDTH - szerokość gadżetu względem szerokości okna.

Oprócz znaczników sterujących położeniem i wielkością gadżetu wpisywanymi do zmiennej Flags w strukturze Gadget istnieją jeszcze inne znaczniki:

GADGHCOMP - zmiana kolorów gadżetu na ich dopełnienia. Gdy włączymy ten gadżet (wciśniemy przycisk myszki podczas gdy kursor znajduje się nad tym gadżetem) to kolory gadżetu zostaną zamienione na kolory odwrotne.

GADGHBOX - gdy gadżet ten jest włączony to zostaje narysowana wokół niego ramka.

GADGHIMAGE - gdy gadżet jest włączony to zostaje wyświetlona grafika określona w zmiennej SelectRender w strukturze Gadget.

GADGHNONE - gdy gadżet jest włączony to nie dzieje się nic.

Jeżeli gadżet ma być wykreślany za pomocą grafiki to ustawiamy także znacznik GADGIMAGE, w innym przypadku (kreślenie ramki lub bez niczego) nie ustawiamy tego znacznika.

SELECTED - ustawiamy ten znacznik jeżeli chcemy aby gadżet został automatycznie włączony kiedy zostaje utworzony. Używamy tego znacznika tylko do gadżetów typu przełączanego. Jeżeli chcemy sprawdzić w trakcie trwania programu czy dany gadżet został włączony to możemy sprawdzić zmienną Flags w strukturze Gadget i w przypadku gdy ten znacznik jest ustawiony to gadżet jest włączony w innym przypadku nie.

GADDISABLED - ustawiamy ten znacznik jeżeli chcemy aby gadżet był niedostępny dla użytkownika. Nie można wtedy dokonywać operacji na tym gadżecie aż do czasu gdy zostanie on ponownie uaktywniony. Aktywacji gadżetu dokonuje się procedurą Intuition.OnGadget() a dezaktywacji procedurą Intuition.OffGadget.

Kolejnym polem zawierającym znaczniki jest pole Activation:

GADGIMMEDIATE - jeżeli chcemy aby nasz program został poinformowany od razu gdy tylko użytkownik włączył dany gadżet to powinniśmy ustawić ten znacznik.

RELVERIFY - ustawiamy ten znacznik jeżeli chcemy aby nasz program otrzymał komunikat jeżeli użytkownik opuści gadżet (na przykład wyłączy gadżet lub puści przycisk myszki po przesunięciu gadżetu typu Proporcjonalny).

FOLLOWMOUSE - ustawiamy ten znacznik jeżeli chcemy aby program otrzymywał komunikaty o przesunięciu myszki podczas gdy dany gadżet jest wybrany przez użytkownika.

TOGGLESELECT - ustawiamy ten znacznik jeżeli chcemy aby gadżet za każdy razem gdy użytkownik go wybierze przełączał się a nie tylko załączał do czasu gdy użytkownik puści przycisk myszki. Następnie możemy sprawdzać pole Flags w strukturze Gadget i jeżeli ustawiony jest znacznik SELECTED to ozancza iż gadżet jest włączony.

BOOLEXTEND - ustawiamy ten znacznik jeżeli nasz gadżet ma dołączoną strukturę BoolInfo. Jeżeli gadżet jest podłączony do okna to możemy ustawić następujące znaczniki w polu Activation aby osiągnąć określone szerokości ramek okna (można tam umieścić gadżet):

RIGHTBORDER - szerokość prawej ramki okna będzie obliczona za pomocą pozycji i szerokości gadżetu.

LEFTBORDER - szerokość lewej ramki okna będzie obliczona za pomocą pozycji i szerokości gadżetu.

TOPBORDER - wysokość górnej ramki okna będzie obliczona za pomocą pozycji i wysokości gadżetu.

BOTTOMBORDER - wysokość dolnej ramki okna będzie obliczona za pomocą pozycji i wysokości gadżetu.

Jeżeli gadżet jest dołączony do requestera to możemy ustawić znacznik ENDGADGET. Requester zostanie zatwierdzony tylko wtedy jeżeli zostanie wciśnięty gadżet z ustawionym tym znacznikiem. Kolejne znaczniki odnoszą się tylko do gadżetów typu Ciąg/Wartość:

STRINGRIGHT - ciąg znaków będzie ustawiany względem prawej ramki w gadżecie.

STRINGCENTER - jeżeli ustawimy ten znacznik to ciąg znaków będzie centrowany. (Standardowo (bez ustawionych znaczników) ciąg jest ustawiany względem lewej ramki.)

LONGINT - ustawiamy ten znacznik jeżeli chcemy aby użytkownik mógł wpisać tylko 32 bitową liczbę dziesiętną całkowitą ze znakiem.

ALTKEYMAP - ustawiamy ten znacznik jeżeli chcemy aby dla gadżetu pobierającego ciąg było używane inne ustawienie klawiatury. Wskaźnik strktury KeyMap wpisujemy wtedy do pola AltKeyMap w strukturze StringInfo.

Ostatnim polem zawierającym znaczniki jest pole GadgetType, które określa typ gadżetu. Musimy ustawić jeden z następujących znaczników:

BOOLGADGET - ustawiamy ten znacznik jeżeli chcemy aby nasz gadżet był typu Przełączany.

STRGADGET - ustawiamy jeżeli nasz gadżet ma być typu Ciąg. Jeżeli chcemy gadżet typu Wartość to musimy także ustawić znacznik LONGINT w polu Activation w strukturze Gadget.

PROPGADGET - ustawiamy ten znacznik jeżeli gadżet ma być typu Proporcjonalny.

Kolejne znaczniki odnoszą się do specjalnych gadżetów (ustawiane są także w polu GadgetType):

GZZGADGET - jeżeli gadżet jest umieszczany w oknie typu GIMMEZEROZERO i ustawimy ten znacznik to gadżet zostanie umieszczony w oknie zewnętrznym i nie uszkodzi żadnych rysunków i tym podobnych rzeczy w oknie wewnętrznym (o oknie typu GIMMEZEROZERO pisaliśmy w Kąciku Początkującego Kodera w numerze 5/92 "64 PLUS 4").

REQGADGET - ustawiamy ten znacznik jeżeli gadżet jest podłączony do requestera.

Gadżet typu Przełączany:

Dla takiego gadżetu ustawiamy odpowiednie znaczniki i wskaźniki oraz istnieje możliwość dołączenia dodatkowej struktury do pola o nazwie SpecialInfo w strukturze Gadget o nazwie BoolInfo.

Struktura BoolInfo00 WORD bi_Flags ; znaczniki dla struktury BoolInfo (aktualnie istnieje tylko znacznik BOOLMASK).

02 APTR bi_Mask ; wskaźnik maski bitowej umożliwiającej określenie obszaru na jakim działa gadżet (na przykład koło). Maska bitowa jest ustalana w słowach a jej szerokość i wysokość muszą być takie same jak szerokość i wysokość gadżetu.

06 LONG bi_Reserved ; ustawiamy na 0 gdyż jest zarezerwowane do przyszłych zastosowań. Znacznikiem dla struktury BoolInfo jest tylko BOOLMASK i należy go zawsze ustawiać jeżeli używamy tej struktury.

Gadżety typu Ciąg i Wartość:

Gadżety typu Ciąg i Wartość są o wiele bardziej skomplikowane od gadżetów przełączanych gdyż musimy zdefiniować dla nich strukturę StringInfo, której wskaźnik wpisujemy do pola SpecialInfo. Gadżety tego typu pozwalają użytkownikowi na wpisywanie ciągów znaków znaków bądź wartości liczbowych. Biblioteka intuition zapewnia całkowitą obsługę tych gadżetów a dla programisty pozostaje jedynie określenie wymiarów gadżetów oraz ich pozycji a także buforów na znaki. Struktura StringInfo00 APTR si_Buffer ; wskaźnik do bufora znaków bądź tekstu zakończonego zerem który się pojawi w gadżecie po jego inicjacji.

04 APTR si_UndoBuffer ; wskaźnik do bufora znaków, który musi być długi jak bufor właściwy (umieszczamy go opcjonalnie i jeżeli nie chcemy to wpisujemy do tego pola zero). Gdy użytkownik wybierze ten gadżet to tekst z bufora właściwego jest kopiowany do tego bufora i jeżeli użytkownik wciśnie kombinację klawiszy AMIGA + Q to zostanie wpisany do gadżetu tekst jaki istniał przed zmianami. Gadżety typu Ciąg i Wartość są inicjowane w ten sposób, że tylko jeden z nich może być aktywny a więc możemy podać ten bufor do wszystkich tych gadżetów użytych w programie jednak musimy pamiętać wtedy iż musi być tak długi jak największy z buforów.

08 WORD si_BufferPos ; wartość określająca pozycję znaku w buforze, który ma być umieszczony pod kursorem.

0a WORD si_MaxChars ; maksymalna ilość znaków, która może trafić do bufora plus zero kończące ten ciąg.

0c WORD si_DispPos ; pozycja pierwszego znaku wyświetalnego w obrębie gadżetu.

0e WORD si_UndoPos ; pozycja w buforze UndoBuffer (ustawiane przez intuition).

10 WORD si_NumChars ; aktualna ilość znaków w buforze (ustawiane przez intuition).

12 WORD si_DispCount ; aktualna ilość wyświetlanych znaków (ustawiane przez intuition)

14 WORD si_CLeft ; lewy róg obszaru do wyświetlania znaków (ustawiane przez intuition).

16 WORD si_CTop ; górny róg obszaru do wyświetlania znaków (ustawiane przez intuition).

18 APTR si_LayerPtr ; wskaźnik struktury Layer dla obszaru wyświetlania (ustawiane przez intuition).

1c LONG si_LongInt ; jeżeli pobieramy wartość liczbową z gadżetu typu Wartość to możemy sprawdzić to pole aby otrzymać tę wartość.

20 APTR si_AltKeyMap ; wskaźnik struktury KeyMap dla danego gadżetu umożliwiającej inne ustawienie klawiatury dla gadżetów.

Inicjacja gadżetu typu Ciąg i Wartość różni się tylko tym, że przy gadżecie typu Wartość musimy wpisać do bufora kod liczby która ma być pokazana przy wejściu do tego gadżetu (najczęściej jest to "0") oraz tym, że musimy ustawić znacznik LongInt w polu Activation w strukturze Gadget.

Gadżet typu Proporcjonalny:

Gadżet typu Proporcjonalny to mały suwak, który może być przemieszczany w pionie, poziomie lub obydwu tych kierunkach na raz ale w obrębie pewnego wyznaczonego obszaru. Może być używany na przykład do pokazania jak wiele danych jest w zbiorze a jaka część jest aktualnie pokazana na ekranie przesunięcie spowoduje pokazanie innej części. Ważną rolę w tego typu gadżecie odgrywa struktura PropInfo:

Struktura PropInfo00 WORD pi_Flags ; znaczniki dla gadżetu typu Proporcjonalny

02 WORD pi_HorizPot ; przesunięcie poziome suwaka (jeżeli znajduje się w skrajnym położeniu to ta wartość wynosi $ffff, jeżeli w przeciwnym to $0000. Każde inne położenie to zmiana tej wartości w tym zakresie (od $ffff do $0000), na przykład jeżeli użytkownik zmieni położenie suwaka o połowę to wartość ta wynosi połowę a więc ($7fff).

04 WORD pi_VertPos ; przesunięcie pionowe suwaka (działanie patrz: pi_HorizPos). pi_HorizPos i pi_VertPos mogą występować jednocześnie.

06 WORD pi_HorizBody ; opisuje jak duży ma być suwak i ile ma wykonywać kroków gdy użytkownik wciśnie przycisk myszki wewnątrz gadżetu a nie na suwaku. Jeżeli na przykład chcemy otrzymywać wartości od 0 do 63 (głośność muzyki) to suwak ma wykonać 64 kroki czyli wielkość suwaka to 1/64 z maksymalnej wielkości suwaka czyli $ffff. HorizBody musimy więc ustawić na wartość wyliczoną z 1/64 * $ffff czyli $3ff. HorizBody opisuje wielkość suwaka a więc także w przypadku gdy wyświetlamy pewne dane jego wielkość określa jaka część danych jest widoczna.

08 WORD pi_VertBody ; działanie takie same jak w przypadku HorizBody za wyjątkiem tego, iż wielkość suwaka jest zmieniana w pionie a suwak oddziaływuje na komórkę VertPot. Następne ze zmiennych są ustawiane przez intuition i programista powinien ustawić je na zero.

0a WORD pi_CWidth ; Szerokość obszaru, w którym ma się poruszać suwak.

0c WORD pi_CHeight ; Wysokość obszaru, w którym ma się poruszać suwak.

0e WORD pi_HPotRes ; wielkość o jaką będzie przesuwany suwak w poziomie.

10 WORD pi_VPotRes ; wielkość o jaką będzie przesuwany suwak w pionie.

12 WORD pi_LeftBorder ; pozycja lewej ramki obszaru, w którym będzie poruszany suwak.

14 WORD pi_TopBorder ; pozycja górnej ramki obszaru, w którym będzie poruszany suwak.

Znaczniki dla struktury PropInfo (znaczniki wpisujemy do pola Flags w tej strukturze).

FREEHORIZ - jeżeli ustawimy ten znacznik to użytkownik będzie mógł poruszać suwakiem w poziomie.

FREEVERT - jeżeli ustawimy ten znacznik to użytkownik będzie mógł poruszać suwakiem w pionie.

AUTOKNOB - ustawiamy ten znacznik jeżeli chcemy aby wielkość suwaka była kontrolowana przez intuition a nie wpisywana przez nas. Gdy ustawimy ten znacznik to na wielkość suwaka wpływają wartości wpisane w zmiennych HorizBody i VertBody ze struktury PropInfo. Gdy inicjujemy taką strukturę z ustawionym znacznikiem AUTOKNOB to powinniśmy podać wskaźnik do struktury Image w zmiennej GadgetRender w strukturze Gadget (nie musimy tej struktury inicjować gdyż intuition się o to zatroszczy).

PROPBORDERLESS - ustawiamy ten znacznik jeżeli nie chcemy ramki wokół obszaru, w którym porusza się suwak.

KNOBHIT - intuition ustawia ten znacznik jeżeli gadżet jest włączony.

BADANIE STANU GADŻETÓW

Gdy już się zdecydowaliśmy jakiego typu i jak wyglądające gadżety chcemy umieścić w naszym programie powinniśmy zdecydować jakie informacje mają przesyłać do programu. Musimy zdecydować czy mają wysyłać informację wtedy gdy użytkownik wcisnął przycisk myszki na gadżecie czy wtedy kiedy go puścił czy w obydwu przypadkach. Całą pracę wykonuje za nas intuition a informację otrzymujemy w postaci znaczników IDCM, o których pisaliśmy w poprzednim numerze Kącika Początkującego Kodera. Gdy już wiemy jekie komunikaty chcemy uzyskać to musimy wykonać następujące kroki:

1. Powinniśmy ustawić odpowiednie znaczniki w polu Activation w strukturze Gadget:

GADGIMMEDIATE - ustawiamy wtedy gdy chcemy aby nasz program otrzymywał komunikaty od razu gdy użytkownik włączy gadżet.

RELVERIFY - ustawiamy wtedy gdy chcemy aby nasz program otrzymał komunikat gdy użytkownk opuści gadżet. Opuszczenie gadżetu następuje tylko wtedy gdy użytkownik puścił przycisk myszki w czasie gdy kursor znajdował się w obszarze wyznaczonym przez gadżet. Gdy zrobi to poza obszarem gadżetu nasz program nie otrzyma żadnego komunikatu.

FOLLOWMOUSE - ustawiamy ten znacznik gdy chcemy aby nasz program otrzymywał komunikaty o przesunięciu myszki gdy gadżet jest załączony. Intuition automatycznie będzie przekazywało wartoście do komunikatu MOUSEMOVE.

2. Powinniśmy przygotować program na otrzymywanie odpowiednich komunikatów jeżeli gadżet mamy przyłączony do okna i przekazać intuition w polu IDCMPFlags w strukturze NewWindow jakie komunikaty ma przesyłać do naszego programu. Z gadżetów możemy otrzymywać następujące komunikaty:

GADGETDOWN - ustawiamy jeżeli gadżety podłączone do okna mają ustawiony znacznik GADGIMMEDIATE. W tym przypadku do portu IDCMP będą przekazywane wszystkie komunikaty wtedy gdy użytkownik włączył jakiś gadżet.

GADGETUP - ustawiamy jażeli w gadżetach podłączonych do okna mamy ustawiony znacznik RELVERIFY. Zostaniemy poinformowani przez intuition o każdum opuszczeniu gadżetu przez użytkownika.

MOUSEMOVE - jeżeli gadżet podłączony do okna ma ustawiony znacznik FOLLOWMOUSE to powinniśmy ustawić ten znacznik.

3. Gdy już zdefiniowaliśmy gadżety i komunikaty to powinniśmy zatroszczyć się o zbieranie i sprawdzanie tych danych. Dokonujemy tego w sposób jak to było opisane w poprzednim odcinku Kącika Początkującego Kodera w artykule traktującym o znacznikach IDCMP. Gdy otrzymaliśmy komunikat GADGETUP lub GADGETDOWN to oznacza, iż użytkownik dokonał operacji na gadżecie a jego adres (oczywiście gadżetu) znajduje się w polu IAddress i teraz możemy dokonać analizy gadżetu aby stwierdzić jakie nastąpiły zmiany.

PROCEDURY NAJCZĘŚCIEJ UŻYWANE PRZY PRACY Z GADŻETAMI

RefreshGadgets() - IntuitionRefreshGadgets (Gadget, Window, Requester) A0 A1 A2

Funkcja ta ma na celu narysowanie wszystkich gadżetów od nowa z listy gadżetów przypisanych dla danego okna czy requestera. Można użyć tej funkcji jeżeli dodaliśmy jakiś nowy gadżet. Wywołujemy ją wtedy tylko dla tego gadżetu. Użyteczna jest też gdy zmieniliśmy wygląd gadżetów bądź coś mogło je zaśmiecić. Funkcji tej używa się także przy odświeżaniu jeżeli dokonujemy tego samodzielnie.

Wejście:

Gadget - wskaźnik do struktury Gadget dla pierwszego gadżetu, od którego ma być rozpoczęte rysowanie. Gadżet wskazany i pozostałe podłączone do niego będą odświeżone.

Window - wskaźnik struktury Window dla okna, w którym ma nastąpić odświeżenie gadżetów.

Requester - wskaźnik struktury Requester, do której ten gadżet jest podłączony. Jeżeli gadżet jest podłączony do requestera to requester musi być wyświetlony przed użyciem tej funkcji! Jeżeli gadżet nie jest podłączony do requestera to wpisujemy tutaj zero.

AddGadget() - IntuitionWynik = AddGadget (Window, Gadget, Pozycja) A0 A1 D0

Funkcja ta dodaje gadżet do listy gadżetów na określonej pozycji. Należy pamiętać aby po wywołaniu tej funkcji wywołać Intuition.Refreshgadgets() aby zobaczyć zmiany, których dokonaliśmy. Można dołożyć lub zabrać kilka gadżetów jednak po tej zmianie należy koniecznie wywołać funkcję Refresh.Gadgets().

Wejście:

Window - wskaźnik struktury Window dla okna, do którego gadżet ma być dodany. Gadget - wskaźnik struktury Gadget gadżetu, który ma być dodany do listy gadżetów.

Pozycja - pozycja w liście gadżetow, na którą gadżet ma być dodany. Jeżeli wynosi ona zero to gadżet zostanie dodany przed wszystkimi gadżetami, natomiast każda konkretna liczba oznacza pozycję za gadżetem o tym numerze. Wartość zbyt duża lub wartość -1 powoduje dodanie gadżetu na końcu listy.Wyjście:Wynik - (długie słowo) - zawiera aktualną pozycję gadżetu w liście gadżetów po tym jak został on dodany.

RemoveGadget() - IntuitionWynik = AddGadget (Window, Gadget) A0 A1

Funkcja ta usuwa gadżet do listy gadżetów dla określonego okna.

Wejście:

Window - wskaźnik struktury Window dla okna, z którego gadżet ma być usuięty.

Gadget - wskaźnik struktury Gadget gadżetu, który ma być usunięty z listy gadżetów.

Wyjście:

Wynik - (długie słowo) - zawiera pozycję gadżetu, który został usunięty lub -1 jeżeli takiego gadżetu nie znaleziono.

OnGadget() - IntuitionOnGadget (Gadget, Window, Requester) A0 A1 A2

Funkcja ta uaktywnia podany gadżet (usuwa ona znacznik GADGDISABLED z pola Flags w strukturze Gadget). Należy pamiętać, że tak długo jak gadżet jest nieaktywny tak długo użytkownik nie może z niego korzystać. Gadżet nieaktywny jast pokryty siateczką ("ghosted").

Wejście:

Window - wskaźnik struktury Window dla okna, z którego gadżet ma uaktywniony.

Gadget - wskaźnik struktury Gadget gadżetu, który ma być uaktywniony.

Requester - wskaźnik struktury Requester dla requestera, do którego gadżet uaktywniany jest podłączony bądź zero jeżeli gadżet jest przypisany do oona.

OnGadget() - IntuitionOnGadget (Gadget, Window, Requester) A0 A1 A2

Funkcja ta deaktywuje podany gadżet (ustawia ona znacznik GADGDISABLED w polu Flags w strukturze Gadget). Należy pamiętać, że tak długo jak gadżet jest nieaktywny tak długo użytkownik nie może w z niego korzystać. Gadżet nieaktywny jast pokryty siateczką ("ghosted").

Wejście:

Window - wskaźnik struktury Window dla okna, z którego gadżet ma zdeaktywowany.

Gadget - wskaźnik struktury Gadget gadżetu, który ma być zdeaktywowany.

Requester - wskaźnik struktury Requester dla requestera, do którego gadżet deaktywowany jest podłączony bądź zero jeżeli gadżet jest przypisany do oona.

ModifyProp() - IntuitionModifyProp (Gadget, Window, Requester, Flags, HorizPot, VertPot, HorizBody, VertBody) A0 A1 A2 D0 D1 D2 D3 D4

Funkcja ta umożliwia modyfikację parametrów gadżetu proporcjonalnego, wszelkich wartości oraz suwaka. Jeżeli na przykład tworzymy requester do wybierania zbiorow to przy czytaniu zbiorów z dysku wielkość suwaka w gadżecie proporcjonalnym musi się zmniejszać. Należy obliczyć o ile, następnie wystarczy wywołać tę funkcję w celu dokonania zmian. Funkcja ta zmieni wszystkie parametry i dokona odświeżenia gadżetu.

Wejście:

Gadget - wskaźnik struktury Gadget dla gadżetu typu Proporcjonalny, dla którego modyfikujemy parametry.

Window - wskaźnik struktury Window dla okna, do którego ten gadżet jest podłączony.

Requester - wskaźnik struktury Requester dla requestera, do którego ten gadżet jest podłączony bądź zero jeżeli gadżet jest przypisany do okna.

Flags - znaczniki zmieniane w strukturze PropInfo. Mogą to być tylko te znaczniki, które są zawarte w tej strukturze.

HorizPot - zawiera aktualną proporcjonalną pozycję poziomą suwaka.

VertPot - zawiera aktualną proporcjonalną pozycję pionową suwaka.

AMIGA

HorizBody - opisuje jak duży ma być suwak i ile ma wykonywać kroków gdy użytkownik wciśnie przycisk myszki wewnątrz gadżetu (ale nie na suwaku).

VertBody - działanie jak w przypadku HorizBody jednak odnosi się do suwaka poruszanego w pionie.Na dzisiaj to już wszystko.

Marcin "Duddie" Dudar

ERRATA - ASMONE

W poprzednim numerze "64 PLUS 4" opublikowaliśmy instrukcję do programu AsmOne. Podczas jej pisania popełniłem kilka błędów, za które chciałbym przeprosić.

W menu Project, w podmenu Preferences opcja Level 7 nie jest wywoływana przez ESC - jak to błędnie napisałem w artykule - lecz jest wywoływana przez specjalny przełącznik wywołujący to przerwanie. Przełącznik możemy zamontować sami (jego schemat postaramy się opublikować w następnym numerze "64 PLUS 4").

W menu Assemble została pominięta opcja ShowSource, służąca do tego by przy wejściu do Debugger'a następowała asemblacja tekstu źródłowego a kursor debugger'a został ustawiony na początku kodu.

W menu Debugger została pominięta opcja Disassemble powodująca, że rozkaz znajdujący się pod kursorem zostaje zdisasemblowany i wydrukowany na dole okna debugger'a.

Za wszystkie błędy i niedociągnięcia przepraszam.

Marcin "Duddie" Dudar

OPIS ZESTAWU PUBLIC DOMAIN PACK

AMIGA

nr 18 (czerwiec '92)

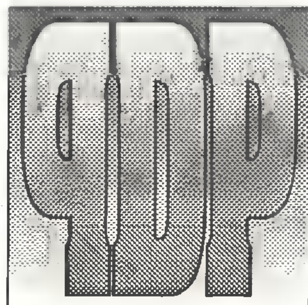
Witam! Jak zwykle po włożeniu dysku do stacji ukazuje nam się stary, sprawdzony boot-block Orcrista z LSD. Jeżeli ktoś nie lubi tego boot-block'a to najlepszym wyjściem będzie skorzystanie z być może niedużej, ale z pewnością składającej się z najlepszych boot-block'ów biblioteki BootX'a. Nie doradzam natomiast początkującemu użytkownikowi zabawy z boot-block'ami z programu VirusExpert v1.81 (wersji 2.0 nie znam), gdyż znajdują się tam nie tylko zwykłe boot-block'i, ale także najwredniejsze wirusy boot-block'owe. Oczywiście dla kogoś, kto jako tako zna Amigę wirus boot-block'owy nie stanowi najmniejszego problemu (chociaż może trochę narozrabiać), ale radzę początkującym bardzo uważać na to co nagrywają na boot'ach. Oto krótki opis jego funkcji Boota:

F1 - Memory On/Off - włączanie bądź wyłączanie fast ram'u.
F2 - Drives On/Off - włączanie lub wyłączanie zewnętrznych stacji dysków.
F3 - Hard reset - zimny start systemu, łacznie z ustawieniem na nowo wszystkich wektorów systemowych (zabija virusy w pamięci).
F4 - Install - nagranie tego boot-block'a na dysk w stacji DF0.
F5 - 60 Hz (ECS) - zmiana częstotliwości wyświetlania obrazu.

Klikamy myszą i cóż nam się ukazuje - nowa ikona dysku, nowe kolory! No dobrze, ale wszystko to jest tylko oprawą wspaniałości znajdujących się na czerwcowym PDP-Amiga!

Po raz pierwszy na Public Domain Pack'u zagościł naprawdę duży program: najnowsza (w chwili pisania tego artykułu) wersja Protrackera nr. 2.2A. Nie dość tego jest to całkowicie kompletna wersja tego programu, wraz ze wszystkimi programami pomocniczymi, helpami, playroutine'ami i innymi. Na PDP znalazły się pilki:

PT2.2A - program główny.
PT2.2A.info - Ikona Protracker'a.
PROP - player do modułów z Protrackera.
PT.TempoCalc.txt - informacja o obliczaniu tempa.
PT-Play.s - playroutine dla asemblerów Seki, lub DevPac'a.
PT-CIAPlay.s - playroutine z taktowaniem CIA.
PT.FileFormat.txt - opis sposobu zapisu ścieżek I sampli do pliku.
PT.Effects.txt - podręczny help o dostępnych efektach specjalnych.
PT.help - help dostępny tylko w czasie działania Protrackera, po użyciu klawisza Help. Uwaga! W żadnym wypadku nie należy edytować tego pliku!
PT.help.txt - normalna (Ascii) wersja helpa.
PT.help.txt.info - ikona pliku PT.help.txt.
PT.config-00 - plik konfiguracyjny.
PT.Update! - plik tekstowy zawierający opis zmian w stosunku do poprzedniej wersji tj. PT2.1A.



PT.Update!.info - ikona pliku PT.Update!.
PT.ReadMe! - zawiera historię Protrackera oraz spis plików.
PT.ReadMe!.info - ikona pliku PT.ReadMe!.

Jeżeli ktoś z was ma swoją ulubioną konfigurację, pieczołowicie ustawioną na Protrackerze 2.1A to może jej także używać na Protrackerze 2.2A pod warunkiem, że zmieni pierwsze pięć znaków tego pliku na PT2.2. Oczywiście wszystkie nowe funkcje będą ustawione na zero. Nowa wersja to może niezbyt dużo nowych rzeczy w stosunku do Protrackera 2.1A , ale zapewniam was, że usunięta została cała góra błędów i teraz praca z Protrackerem to czysta przyjemność. Od autora programu czyli "Spread the program not the words!" :

Protracker jest programem Public Domain, może być kopiowany bez ograniczeń, ale nie wolno za ten program żądać zapłaty wyższej niż wynoszą koszty kopiowania. W miarę możliwości należy do Protrackera dołączać wszystkie pliki, z którymi był standardowo. Bardzo ważne jest podczas robienia własnego dysku z Protrackerem (ale przecież nie ma takiej potrzeby!) jest skopiowanie powerpacker.library oraz najważniejszego dla Protrackera pliku - PT.help. Jest to minimum dla poprawnego działania programu. Jeżeli znajdziesz jakieś błędy to napisz pod ten adres: Peter "CRAYON" Hanning Vinterstigen 12 14440 Ronninge Sweden lub pod ten: Anders "DOLPHIN" Ramsay Fornhojdsvagen 6 II 15158 Sodertalje Sweden. Muzykom życzę taaaakich modułów stworzonych przy pomocy Protrackera!

Wow! Nowe wielkie party w Austrii! Intro "Party" to właśnie zaproszenie na tę imprezę. Nie jest zachwycające, ale cóż - to tylko zaproszenie. Party jest organizowane przez dwie grupy - amigowską Energy oraz grupę ze sceny C-64 czyli Cosmos Designs. Party rozpoczyna się w czwartek, 3 września o godzinie dziesiątej a kończy w sobotę, 5 września o godzinie osiemnastej. Odbędzie się w miejscowości Gleisdorf na południe od Graz w Austrii. Zapamiętaj jak się dostać na party: po wysiadce z pociągu w Gleisdorf'ie należy opuścić stację i wsiąść do autobusu (na szczęście jest tylko jeden), autobus opuszczamy na drugim przystanku (na Hauptplatz) i co ukazuje się naszym oczom (no, powinno się ukazać) - klika znaków wskazujących jak dostać się na party, które będzie ok. 50 metrów dalej. Jeżeli ktoś chce przyjechać samochodem to radzę zaopatrzyć się w mapkę u organizatorów. Wystarczy skontaktować się telefonicznie lub listownie. Adres i telefon podam na końcu. Podobno będą dwa pokoje sypialne więc jeśli chcesz nocować to zabierz ze sobą śpiwór. Dla amatorów wygód jest możliwość zarezerwowania miejsca w hotelu. Na terenie Party będzie również kawiarnia, w której będzie można coś wypić i zjeść. Należy się spodziewać jedzenia w rodzaju Pizzy, frytek, Hot dogów etc. Oczywiście będzie Competition, nagrody są wysokie a prawdopodobnie będą jeszcze wyższe (zależnie od liczby uczestników):

Amiga demo:	Commodore 64 demo:
1. 7000 szylingów	1. 7000 szylingów
2. 3000 szylingów	2. 3000 szylingów
3. 1000 szylingów	3. 1000 szylingów.

Jeżeli na Party będzie ponad 200 osób (a raczej będzie!) to zaplanowane jest także graphics oraz sound competition. Odbędzie się również Kick Off competition - to dla graczy! Będzie możliwość zakupu koszulek T-Shirt z Party (ok. 90 szylingów) wejście kosztuje 200 szylingów. Adres i telefon pod jaki trzeba się kontaktować w sprawie autokaru, hotelu, lub mapy, a także co najważniejsze rezerwacji stolika. Energy Postlagernd A-8160 Weiz Austria lub telefonicznie (+43) 317-230-575

Wracając do naszego Public Domain Pack'u to niestety niewiele więcej się zmieściło na dysku (sam Protracker z pomocniczymi programami to ok. 400KB...) Na pocieszenie kilka iff'ów, oraz obiecany program "Wykresy" w wersji tekstowej i "AMOS'owej".

Public Domain Pack

nr 19 (lipiec '92)

Opis zestawu zacznijmy od rozrywki.

Proponowaną grą planszową jest Otello (po angielsku othello). Niestety pozory mylą i po naciśnięciu Othello.DOC okazuje się, że na dysku nie ma instrukcji do tej gry, postaram się więc opisać jej zasady. Gra polega na przemiennym ustawianiu na planszy czarnych lub białych klocków. Plansza ma wymiary 8 na 8 pól. Każdy z graczy posiada swój kolor żetonów, czarny lub biały, na początku obaj gracze otrzymują po dwa klocki umieszczone w środku planszy na krzyż z pionkami przeciwnika. Po dostawieniu pionka przez gracza wszystkie żetony wroga, znajdujące się w wyniku tego ruchu pomiędzy naszymi żetonami zmieniają kolor na nasz, przy czym pionki przeciwnika mogą się również zawierać w naszych po skosie a nie tylko w pionowej lub poziomej linii prostej. Gracz ma prawo dostawić pionek tylko wtedy, gdy ten ruch spowoduje zmianę co najmniej jednego żetonu przeciwnika, w przeciwnym wypadku należy zrezygnować z ruchu korzystając z gadżetu "Miss Go". Wygrywa ten z graczy, który po zapełnieniu ostatniego pola posiada więcej żetonów swojego koloru na planszy. Gra kończy się również wtedy gdy na planszy pozostaną tylko żetony jednego koloru. Oczywiście w tym wypadku wygrywa ten, którego żetony pozostały na stole.

Ruch zawsze zaczynają żetony białe (kolor jest losowany). Aby zacząć grę od początku należy użyć gadżetu "Restart". Komputer okazuje się zadziwiająco dobry w rozgrywce, aby go pokonać należy rzeczywiście się przyłożyć. Program Othello ma dobrą cechę, mianowicie pracuje w eleganckim okienku na ekranie workbencha. Wobec tego można pograć sobie w tę gierkę podczas pakowania, formatowania bądź optymalizowania dyskietki itd.

Dla koderów niewątpliwie interesującą pozycją będzie wersja źródłowa programu w postaci kodu maszynowego.

Po tym giercowym wstępie czas na coś poważniejszego. Zabierzmy się za użytki. Pierwszy z nich to QCal. Program ten jest podręcznym kalendarzem do 9999 roku. Główną jego zaletą jest wygoda obsługi oraz fakt, iż do każdego dnia można dodać komentarz długości ok. 96 znaków. Oczywiście można zapisać wszystkie komentarze na dysku komendą SAVE. Podczas używania tego programu otwierają się przed nami wielkie możliwości, możemy np. cofając się w czasie znaleźć dzień tygodnia swojego urodzenia, można zapisywać ważne imprezy, klasówki, spotkania. Program odczytuje datę z zegara systemowego, zaraz po uruchmieniu, lub na życzenie po wciśnięciu "Today". Next oraz Prev służą do odszukania następnego - next, lub poprzedniego - prev zapisu. Save jak już wspomniałem służy do zapisywania notatek na dysk, Del służy do ich kasowania, natomiast UWAGA "Purge" wymazuje WSZYSTKIE notatki, więc radzę uważać by zbyt pochopnie nie zniszczyć wszystkich zapisków. Także do tego programu są wszystkie "źródłówki", ale tym razem w języku C.

Następnym małym użytkiem jest THIPerformance v 2.8 - program ten służy do dokładnego przetestowania każdej stacji dysków lub dysku twardego pod względem szybkości. Test jest zarówno dokładny jak i długi. Testowanie mojej stacji wewnętrznej zajęło chyba z pół godziny, wyniki możecie obejrzeć, są zapisane w formacie pliku IFF o nazwie "Przykład_test".

Jak używać THIPerf'a? O, to banalne! Wystarczy wpisać nazwę urządzenia w okienku "device", może to być np. df0:, df1:, dh0:, dh3: czy coś w tym rodzaju. Następnie należy kliknąć na "Test" i po prostu cierpliwie czekać, a po teście ewentualnie można jeszcze zapisać same wyniki - "Save", lub cały obraz ekranu - "SaveIFF". Wyniki ładujemy za pomocą "Load", nie radzę tylko robić tego z plikiem IFF (tym, który tworzymy za pomocą "SaveIFF").

AMIGA

Kolejny użytek to Fansi - program służący do rysowania w popularnym w BBS'ach formacie Ansi. Program ma dużo opcji bardzo pomagających w rysowaniu np. plansz tytułowych do BBS'u. Posiada tzw. Tool-box na dole ekranu, gdzie możemy wybrać znak, którym mamy zamiar rysować. Możliwe jest rysowanie linii, prostokątów, wypełnionych prostokątów, które składają się tylko z wybranego znaku. Dostępne są rozmaite tryby rysowania (z klawiatury, myszą i inne). Możliwe jest kopiowanie bloków, przenoszenie, usuwanie. Gumka działa tak samo jak w Deluxe Paint'cie - wraz z prawym klawiszem myszki. Podsumowując rysowanie programem Fansi jest bardzo wygodne, rysuje się szybko i bez jakichkolwiek problemów. Jest on godny polecenia osobom, które potrzebują plansz w formacie ansi.

Ostatni już użytek to D-Copy v 3.0 - książe wśród programów kopiujących. Ma mnóstwo zalet i prawie żadnych wad. Wystrój graficzny jest naprawdę na poziomie, a nawet powiedziałbym, że znacznie go przewyższa i jest po prostu wspaniały. Dostępne są dwa nibblery, dwa rodzaje kopiowania doscopy, program posiada także wbydowany checkdisk oraz speedtest. Wszystkie gadżety są trójwymiarowe co bardzo podnosi walory estetyczne. Jeżeli na dysku wystąpi jakiś błąd to nie potrzeba sięgać do jakichś ściągawek i zastanawiać się np. "co ma znaczyć ta czwórka na 53 ścieżce". O nie, w takim wypadku kulturalnie wciskamy ERR następnie niewiadomą ścieżkę i już wszystko się nam natychmiast rozjaśnia. Bardzo łatwe jest też zaznaczanie początku i końca obszaru do kopiowania. Najeżdżamy na track-display (podwójna siatka kwadracików) i lewym klawiszem myszy wybieramy scieżkę początkową a prawym dokonujemy wyboru końcowej. Oczywiście istnieje opcja KILL, dzięki której na Amidze 1MB można kopiować całe dyskietki. Dopóki nie użyjemy tej opcji możemy spokojnie wyjść z programu nie czyniąc systemowi żadnej szkody. Należy też zauważyć, iż po wykonaniu operacji KILL nie psuje się wygląd ekranu, tak jak to ma miejsce w "nieśmiertelnym" X-Copy.

Ostatnią pozycją na lipcowym Public Domain Pack'u jest Glass - małe interko zrobione przez uznaną grupę Anarchy. Intro mimo, że składane ze starych procedurek jest naprawdę na poziomie. Na ekranie widzimy obracające się dookoła "szklane" sześciany jeden w drugim, a pod spodem znajduje się ciekawy logos (Podobno też bardzo stary!). Pod spodem jak zwykle jest scroll, ale nie byle jaki - fonty są naprawdę bardzo ładne.

To już naprawdę koniec. Na otarcie łez pozostawiłem kilka niezłych ray-trace'owych iff'ów, oraz bardzo ładną (moim skromnym zdaniem) animację kuli ziemskiej. Aby przerwać oglądanie animacji i zwolnić pamięć należy wcisnąć prawy klawisz myszy. Klawisz lewy służy do zatrzymywania i ponownego startowania animacji. Animacja działa na standardowym screenie custom i mimo, że nie widać kursora można zciągnąć ekran na dół lub użyć jego gadżetów (wiem, że ich nie widać, ale trzeba celować tam gdzie normalnie powinny być). Można też używać kombinacji lewa amiga+N lub M.

Michał "Amber" Gosztyła

AMIGA

AMOS THE CREATOR

- czyli piszemy własną grę! Cz. 1

Począwszy od nieniejszego numeru 64+4&Amiga będziemy zamieszczać kolejne odcinki kursu świetnego języka programowania jakim z pewnością jest AMOS. Język ten naprawdę trudno jest nazwać zwykłym basic'iem ponieważ posiada rozkazy umożliwiające programowanie strukturalne jak też wiele rodzajów pętli (w sumie ponad pół tysiąca komend!) Jest przy tym bardzo szybki, wykorzystuje całą moc Amigi, możliwe jest układanie własnych Copper List'ów itp. W tym numerze zapoznacie się z obsługą programu i podstawowymi wiadomościami na temat zmiennych używanych w AMOS'ie. Stopniowo będziemy zaznajamiać się z bardziej skomplikowanymi zagadnieniami.

Na początku podaję nazwę komendy wyświetloną na ekranie, w nawiasie znajduje się kombinacja klawiszy której możemy użyć zamiast posługiwać się myszką.

RUN (F1) - uruchamia program aktualnie znajdujący się w buforze. Uwaga, AMOS pozwala na jednoczesną pracę nad wieloma programami, bedzie to dokładnie omówione później.

TEST (F2) - sprawdza poprawność składni programu w buforze (automatycznie wykonywany podczas RUN).

INDENT (F3) - służy do porządkowania tekstu.

BLOCKS MENU (F4) - przełącza na Blocks Menu, służące do operacji na blokach.

SEARCH MENU (F5) - tak samo jak w przypadku powyżej, z tym że wyświetla Search Menu.

RUN OTHER (F6) - umożliwia uruchomienie programu z pamięci (nie z bufora!). Jak już wspomniałem AMOS umożliwia pracę nad wiecej niż jednym programem w pamięci, przy czym możemy w danym momencie edytować jeden z nich.

EDIT OTHER (F7) - ładuje do bufora (edytora) program z pamięci.

OVERWRITE (F8) - zmienia tryb wprowadzania znaków.

FOLD/UNFOLD (F9) - służy do zwijania i rozwijania procedur. Jest to bardzo przydatna opcja, która pozwala "połapać się" w bardziej rozbudowanym programie.

SYSTEM MENU: Do tego menu możemy łatwo się dostać używając myszy - wystarczy tylko nacisnąć prawy klawisz myszy i trzymając go wybrać lewym klawiszem którąś z opcji.

LOAD (SHIFT+F1 lub AMIGA+L) - załadowanie programu źródłowego w AMOS'ie do edytora.

SAVE (SHIFT+F2 lub AMIGA+S) - zapisuje program z edytora.

SAVE AS (SHIFT+F3) - zapisuje program, prosząc o podanie nazwy.

MERGE (SHIFT+F4) - dołącza program z dysku do edytora w miejscu położenia kursora.

MERGE ASCII (SHIFT+F5) - dołącza program zapisany w standarcie ASCII do bufora.

AC.NEW/LOAD (SHIFT+F6) - ładuje nowy zestaw programów pomocniczych.

LOAD OTHERS (SHIFT+F7) - ładuje program pomocniczy (może to być Twój program) do pamięci.

NEW OTHER (SHIFT+F8) - usuwa z pamięci jeden z programów pomocniczych.

QUIT (SHIFT+F10) - koniec pracy z programem.

BLOCKS MENU:

Blocks Menu służy do wszelkich operacji na blokach w edytorze, można je wywołać za pomocą klawisza CTRL.

BLOCK START (CTRL+B lub F1) - zaznaczenie początku bloku (w miejscu gdzie znajduje się kursor). Blok można zaznaczyć także za pomocą myszy używając prawego klawisza myszy.

BLOCK END (CTRL+E lub F6) - zaznaczenie końca bloku.

BLOCK CUT (CTRL+C lub F2) - powoduje skasowanie zaznaczonego bloku.

BLOCK PASTE (CTRL+P lub F7) - czyli skopiowanie zaznaczonego bloku w miejsce gdzie znajduje się kursor.

BLOCK MOVE (CTRL+M lub F3) - przeniesienie zaznaczonego bloku na miejsce położenia kursora (stary blok zostanie skasowany).

BLOCK STORE (CTRL+S lub F8) - czyli skopiowanie bloku do pamięci.

BLOCK HIDE (CTRL+H lub F4) - wyłączenie podświetlenia bloku.

BLOCK SAVE (CTRL+F9) - zapisuje blok na dysku, będzie on traktowany jako program.

SAVE ASCII (CTRL+F5) - zapisuje zaznaczony blok na dysk w postaci pliku ASCII.

BLOCK PRINT (CTRL+F10) - wydrukowanie bloku.

SEARCH MENU:

Search Menu przyda nam się gdy musimy znaleźć jakiś fragment programu lub wtedy gdy np. trzeba zamienić zmienną N na N# w całym programie itp.

FIND (ALT+F1) - wyszukuje w programie ciąg tekstowy (do 32 znaków długości).

FIND NEXT (ALT+F2) - szyka następnego ciągu po odnalezionym, korzystając z tego samego wzorca.

FIND TOP (ALT+F3) - działa tak jak FIND, z tym że poszukiwania zaczyna od początku programu.

REPLACE (ALT+F4) - oczekuje wprowadzenia ciągu znaków i ciągu jaki ma go zastąpić. Jeżeli przedtem użyłeś FIND, to wystarczy że wprowadzisz ciąg do wstawienia (AMOS będzie szukał według starego wzorca).

REPLACE ALL (ALT+F5) - działa tak jak REPLACE ale nie zatrzymuje się i dokonuje zmian w całym programie.

LOW<>UP (ALT+F6) - zmienia kierunek poszukiwań (góra/dół). OPEN ALL (ALT+F7) - bardzo przydatna opcja. Służy do rozwijania wszystkich procedur. Dzięki temu nie trzeba czasem mozolić się z FOLD/UNFOLD.

CLOSE ALL (ALT+F8) - zwija wszystkie procedury.

SET TEXT B(uffer) (ALT+F9) - prosi o podanie nowej ilości pamięci na program źródłowy. Uwaga! Powoduje skasowanie aktualnego programu (AMOS zapyta czy naprawdę jesteś na to zdecydowany).

SET TAB (ALT+F10) - pozwala ustawić liczbę spacji na klawiszu TAB.

KLAWISZE PRZYDATNE PODCZAS EDYCJI PROGRAMU:

Backspace - kasuje jeden znak na lewo od kursora.

Del - kasuje znak, na którym aktualnie jest kursor.

Return - zatwierdzenie linii/przeniesienie ciągu będącego na prawo od kursora o linię w dół.

SHIFT+Backspace - skasowanie linii, na której jest kursor.

CTRL+Q - kasuje część linii na prawo od kursora.

CTRL+U - tzw. undo, przywraca skasowaną linię lub jej część.

CTRL+Y - kasuje linię poniżej kursora.

CTRL+I - wstawia wolną linię tam gdzie znajduje się kursor.

Klawisze strzałek - obsługa kursora na ogólnie znanych zasadach.

SHIFT+lewo, prawo - przesuniecie kursora o jedno słowo.

SHIFT+góra, dół - przesunięcie kursora na początek lub koniec strony.

CTRL+góra, dół - przesunięcie kursora do poprzedniej lub następnej strony.

SHIFT+CTRL+góra, dół - przeniesienie kursora na początek lub koniec całego tekstu.

AMIGA+kierunek - przesuwanie tekstu, przy pozostawieniu kursora na miejscu.

TAB - przesuwa linię do następnej pozycji tabulatora.

SHIFT+TAB - to samo, tylko przesuwa linię do poprzedniej pozycji tabulatora.

CTRL+TAB - ustala długość pozycji tabulatora.

KLAWISZE OBSŁUGI PROGRAMÓW W PAMIĘCI:

AMIGA+P - przeniesienie programu z bufora do pamięci.

AMIGA+F - zamiana programu z bufora (edytora) na program znajdujący się aktualnie w pamieci.

AMIGA+T - wybór kolejnego programu z pamięci.

ZNACZNIKI:

CTRL+SHIFT+(0-9) - wstawienie znacznika w pozycji kursora.

CTRL+(0-9) - ustawienie kursora na wybranym (0-9) znaczniku.

CTRL+C - w czasie działania programu powoduje jego przerwanie (break).

Gdy wciśniemy któryś z klawiszy AMIGA to pojawi się dodatkowe menu naczęściej używanych komend, które możemy wprowadzić jednym z klawiszy funkcyjnych. Komendy te można zmieniać za pomocą instrukcji KEY$(x)=nasza_komenda. X to numer klawisza funkcyjnego (od 1 do 10 dla AMIGI lewej a od 11 do 20 dla prawej). Istnieją pewne znaki specjalne używane przez KEY$. Są to: '(apostrof) - czyli "symulacja" klawisza Return '(ALT+apostrof) - powoduje zignorowanie dalszego tekstu np. Key$(1)="Print'" - po wciśnięciu klawisza F1, komputer wykona komendę Print albo Key$(2)="Cls 0'to jest nieważne" - po wciśnięciu klawisza F2 komputer wyczyści ekrana a resztę komendy zignoruje.

Bezpośrednio pod menu znajduje się linia informująca nas o kilku rzeczach, które warto wiedzieć. LINIA:

^I L=1 C=1 Text=20000 Chip=315000 Fast=457000 Edit:Noname

Po kolei od lewej do prawej dane te oznaczają:

1. Tryb wprowadzania znaków OVERWRITE/-INSERT.
2. Linia, w której aktualnie znajduje się kursor.
3. Kolumna, w której znajduje się kursor.
4. Text - to ilość pamięci pozostałej jeszcze na edycję wersji źródłowej programu. W razie potrzeby można ją zwiększyć komendą Set Buffer znajdującą się w Search Menu.
5. Chip - podaje ilość wolnej pamięci Chip w kilobajtach.
6. Fast - to samo dla pamięci Fast.
7. Nazwa programu.

Wielką zaletą AMOS'a jest fakt, iż posiadamy osobne okno edycji i działania programu. Do przełączania pomiędzy tymi trybami służy klawisz Escape. Gdy program skończy swoje działanie lub przerwiemy jego pracę za pomocą kombinacji CTRL+C automatycznie przecho-

AMIGA

dzimy do trybu natychmiastowego pracy AMOS'a. Tryb ten służy do testowania komend, sprawdzania zawartości zmiennych itp. W tle widać ekran otwarty przez nasz program (o otwieraniu ekranów graficznych w dowolnej rozdzielczości napiszę później) na dole jest pasek na którym wpisujemy komendy. Efekty ich działania przeważnie widać na otwartym ekranie (zależy od rodzaju komendy, oczywiście może się zdarzyć, że wszystkie ekrany graficzne są zamknięte). Położenie naszego paska możemy zmieniać za pomocą klawiszy góra/dół, natomiast jego grubość zmieniamy za pomocą kombinacji SHIFT+góra/dół. Tryb bezpośredni posiada także swój bufor na wprowadzone komendy. Dzięki temu można je powtarzać korzystając z klawiszy funkcyjnych.

Po tym wcale nie krótkim, ale niezbędnym wstępie możemy zacząć zaznajamiać się z możliwościami, trybami pracy i komendami AMOS'a.

Na początek spróbuj wpisać np:

print "To działa!"

Jak zapewne zauważyłeś AMOS zmienił nieco twój program, teraz wygląda on tak:

Print "To działa!"

Jak widzisz AMOS zmienił pierwszą literę rozkazu na dużą. Jest to kolejna wielka zaleta tego języka, gdyż po pierwsze program zawsze wygląda bardzo schludnie i przejrzyście, a po drugie szybko się zorientujesz gdy np. pomylisz literę w komendzie. Po prostu po wpisaniu np:

Prnt "Ooo! To nie będzie działać!"

AMOS wyświetli coś takiego:

PRNT "Ooo! To nie będzie działać!"

co ewidentnie rzuca się w oczy i od razu taki błąd zauważymy.

Niestety nie mogę tutaj opisywać całej podstawowej składni języka BASIC, zakładam, że czytający ten artykuł zna przynajmniej podstawy tego języka. W cyklu tych artykułów postaram się jedynie omówić szczególne cechy i komendy AMOS'a, co w połączeniu z podstawową wiedzą powinno każdemu wystarczyć do efektwnego programowania w tym języku.

Skoro już -mniej więcej wiemy jak działa AMOS od strony edycyjnej, możemy zacząć zapoznawać się ze zmiennymi występującymi w AMOS'ie.

Podstawowym typem zmiennej w AMOS'ie jest zmienna całkowita. Za zmienną całkowitą AMOS uważa każdy ciąg liter i cyfr, którego pierwszym wyrazem jest litera. Może to być np: X albo ilość, albo M4FG lub cokolwiek innego zgodnego z powyższą zasadą. Nazwa zmiennej nie może być komendą AMOS'a (co jest chyba oczywiste). Każda zmienna całkowita jest zapisana w czterech bajtach więc może zawierać się w przedziale od - 147.483.648 do 147.483.648.

Drugim typem zmiennych są zmienne rzeczywiste. Aby otrzymać zmienną rzeczywistą, należy na końcu nazwy zmiennej dopisać znak #. Nazwa zmiennej rzeczywistej może wyglądać np. tak: F# albo Jaś#, albo F333k# itp. Zmienne całkowite są także zapisywane w czterech bajtach, a ich granice to 1E-19 i 1E18, z dokładnością do siedmiu cyfr po przecinku. Z tych dwóch zmiennych zaleca się w miarę możliwości używać zmiennych całkowitych, gdyż operacje na nich są o wiele szybsze. Należy jednak uważać na zmienne całkowite, gdyż wpisanie np:

X=2 .69:Print X

spowoduje wyświetlenie: 2. Może to być przyczyną bardzo trudnych do wykrycia błędów. Gdy np. zwiększamy liczbę rzeczywistą o dajmy na to 4 to lepiej jest wpisać

X#=X#+4.0 niż X#=X#+4

gdyż komputer w drugim przypadku i tak będzie musiał zamienić liczbę całkowitą na rzeczywistą, a to zabierze mu trochę czasu. Liczby w AMOS'ie można zapisywać w dowolny sposób, np. liczbę 100 dziesiętnie można zapisać jako $64 lub %1100100. Jest to czasem bardzo wygodne gdyż nie musimy wcale przeliczać danych na inne systemy liczenia. Musimy tylko pamiętać, że program zawsze będzie wyświetlać wyniki w systemie dziesiętnym i jeżeli napiszemy np. Print $ff to program po uruchomieniu wyświetli nam: 255.

To już wszystko w tym odcinku kursu AMOS'a, w następnym poznamy procedury, operacje na łańcuchach, operacje arytmetyczne i kilka specyficznych komend.

Na zachętę dla zajęcia się AMOS'em przedstawiam króciótki program, który rysuje na ekranie prostego fraktala. Namawiam do jego analizy.

```
Screen Open 0,640,200,4,Hires
Cls 0
Curs Off
Flash Off
Palette 0,$FFF,$999,$777
X1=0 : Y1=199
X2=639 : Y2=199
X3=320 : Y3=0
X=300 : Y=100
Draw X1,Y1 To X2,Y2
Draw X1,Y1 To X3,Y3
Draw X3,Y3 To X2,Y2
For I=0 To 30000
N=Rnd(2)
If N=0 Then Ink 1 : X=X+(X1-X)/2 : Y=Y+(Y1-Y)/2
If N=1 Then Ink 2 : X=X+(X2-X)/2 : Y=Y+(Y2-Y)/2
If N=2 Then Ink 3 : X=X+(X3-X)/2 : Y=Y+(Y3-Y)/2
Plot X,Y
Next
```

Michał "Amber" Gosztyła

REQTOOLS LIBRARY CZ.2

AMIGA

W dzisiejszym odcinku zmagań z ReqTools Library przedstawiamy kolejne funkcje dostępne w tej bibliotece

rtEZRequestA

Wartość = rtEZRequestA (Tekst, GadTekst, ReqInfo, Argumenty, TagList)

A1 A2 A3 A4 A0

Funkcja ta wywołuje requester dla użytkownika i oczekuje na odpowiedź poprzez wybranie przez użytkownika jednego z gadżetów dostępnych w requesterze lub klawisza odpowiadającego danemu gadżetowi. Jeżeli odpowiedź jest pozytywna funkcja ta zwraca wartość dodatnią a jeżeli negatywna to wartość zerową. Funkcja ta może zwracać także znaczniki IDCMP jeżeli tego zarządamy ustawiając odpowiednie rozkazy w liście rozkazów (TagList). Wartość zwracana przez tę funkcje może także odpowiadać kolejnym gadżetom (jeżeli jest ich więcej umieszczonych w requesterze).

GadTekst to tekst zakończony zerem określający teksty odpowiedzi pojawiające się w gadżetach. Odpowiedzi te oddzielamy znakiem pionowej kreski ("|") i tak dla przykładu może to wyglądać następująco: "Tak|Nie", "1|2|3|4" itp. Odpowiedzi w tekście powinny być wpisane w kolejności w jakiej będą się pojawiały gadżety w requesterze od strony lewej do prawej. Nie istnieje limit odpowiedzi a jest jedynie on ograniczony szerokością requestera jaki może się pojawić na ekranie. Tekst może zawierać znak returna ("nowej linii" o kodzie $a) co wymusza na requesterze utworzenie kolejnej linii tekstu. Można także używać kodów formatujących znanych doskonale użytkownikom języka C, amianowicie kodów typu: %s, %ld itp. Kodów tych możemy także używać w gadżetach umieszczając je w tekście odpowiedzi GadTekst. Argumenty umieszczone od wskaźnika "Argumenty" są najpierw pobierane dla tekstu Tekst a następnie dla GadTekst. Formatowanie jest dokonywane za pomocą procedury z Exec Library o nazwie RawDoFmt(), która jest zorientowana na słowa a więc jeżeli chcemy wyświetlić liczbę 32 bitową to zamiast %d (oznaczającego liczbę dzisiętną) musimy wpisać %ld (oznaczające właśnie liczbę dziesiętną zapisaną na długim słowie).

Na requester można odpowiadać za pomocą klawiszy "Y" i "V" jeżeli chcemy udzielić odpowiedzi pozytywnej lub "ESC", "N", "B", "R" jeżeli chcemy udzielić odpowiedzi negatywnej. Jeżeli nie jest ustawiony znacznik EZREQF_NORETURNKEY to klawisz RETURN działa jako zatwierdzenie odpowiedzi pozytywnej. Odpowiedź zatwierdzana przez RETURN może być dowolnie wybrana z listy odpowiedzi a jej numer umieszczony w RTEZ_DefaultResponse. Odpowiedź zatwierdzana przez RETURN będzie wydrukaowana jako wytłuszczona.

Jeśli chcemy pokazać requester zatwierdzający akcję destrukcyjną (na przykład kasowanie zbiorów) to możemy ograniczyć wykorzystanie klawiatury do zatwierdzania requestera do dwóch kombinacji klawiszy jak to ma miejsce w requesterach systemowych czyli AMIGA + V dla zatwierdzenia i AMIGA + B dla zaniechania poprzez ustawienie znacznika EZREQF_LAMIGAQUAL. Należy pamiętać, iż klawisze RETURN i ESC będą nadal aktywne. Klawisz RETURN można zablokować poprzez ustawienie znacznika EZREQF_NORETURNKEY. Klawisza ESC zablokować nie można!

Możemy otworzyć requester bez odpowiedzi to jest poprzez wpisane zera do GadTekst. W takiej sytuacji należy ustawić pewne znaczniki IDCMP przekazując je do requestera za pomocą rozkazu RT_IDCMPFlags z listy rozkazów co pozwoli użytkownikowi na opuszczenie requestera bez potrzeby udzielania odpowiedzi przez wybranie danego gadżetu. Możemy na przykład ustawić znacznik IDCMP MOUSEBUTTONS co spowoduje opuszczenie requestera po wciśnięciu przycisku myszki w jego obrębie. Drugą metodą obsługi takiego requestera jest tak zwany ReqHandler pozwalający na powrót do programu po otwarciu requestera i zamknięcie programowe.

Wejście:

Tekst - wskaźnik tekstu zakończonego zerem, w którym mogą występować kody formatujące tekst, który będzie wyświetlony jako tekst główny w requesterze.

GadTekst - wskaźnik tekstu dla gadżetów z odpowiedziami, ułożonymi od strony lewej do prawej oraz podzielonymi znakiem pionowej kreski ("|") i zakończonymi zerem.

Argumenty - wskaźnik do tablicy zawierającej argumenty dla formatowania ciągów znaków Tekst i GadTekst.

ReqInfo - wskaźnik struktury ReqInfo przydzielonej za pomocą funkcji rtAllocRequest() służącej dla lepszego kontrolowania requesterów lub zero jeżeli tej struktury nie podajemy.

TagList - wskaźnik tablicy rozkazów dla requestera.

Dostępne rozkazy dla rtEZRequestA:

RT_Window - jako parametr podajemy wskaźnik

struktury Window dla okna, które będzie użyte aby odnaleźć ekran, na którym ma być wyświetlony requester. Jeżeli program jest zadaniem a nie procesem to musimy koniecznie podać ten rozkaz gdyż zadania (w przeciwieństwie do procesów) nie mają pr_WindowPtr.

RT_IDCMPFlags - jako parametr podajemy długie słowo, które zawiera znaczniki IDCMP, przy napotkaniu których requester ma powrócić do programu.

RT_ReqPos - jako parametr podajemy długie słowo zawierające jeden z następujących znaczników:

REQPOS_POINTER - requester będzie się pojawiał tam gdzie znajduje się kursor myszki.

REQPOS_CENTERSCR - requester będzie centrowany na ekranie.

REQPOS_CENTERWINDOW - requester będzie centrowany w oknie (działa tylko wtedy jeżeli pr_WindowPtr jest -właściwy lub gdy użyliśmy rozkazu RT_Window). Jeżeli RT_Window jest równe zero to requester bedzie centrowany na ekranie.

REQPOS_TOPLEFTSCR - requester będzie się pojawiał w lewym górnym rogu ekranu.

REQPOS_TOPLEFTWIN - requester będzie się pojawiał w lewym górnym rogu okna (jeżeli pr_WindowPtr jest właściwy lub gdy użyliśmy rozkazu RT_Window)

RT_LeftOffset - jako parametr podajemy długie słowo zawierające przesuniecie od lewej krawędzi wyznaczonej przez rozkaz RT_ReqPos (nie odnosi się to do requesterów, które mają miejsce pojawiania się zależne od pozycji kursora myszki - RT_ReqPos jest określone jako REQPOS_POINTER).

RT_TopOffset - jako parametr podajemy długie słowo zawierające przesuniecie od górnej krawędzi wyznaczonej przez rozkaz RT_ReqPos (nie odnosi się to do requesterów, które mają miejsce pojawiania się zależne od pozycji kursora myszki - RT_ReqPos jest określone jako REQPOS_POINTER).

RT_Screen - jako parametr podajemy wskaźnik struktury Screen dla ekranu, na którym ma się pojawić -requester.

RT_ReqHandler - używając tego rozkazu możemy wystartować asynchroniczny requester. Jako parametr podajemy wskaźnik struktury rtHandlerInfo. Requester zainicjuje ten wskaźnik i powróci do programu po jego normalnej inicjalizacji. Jeżeli wartość zwrócona do programu nie jest taka jakiej moglibyśmy się spodziewwać, i jeżeli nie jest równa wartości CALL_HANDLER to znaczy że wystąpił błąd i należy ustawiać parametry dla requestera od początku. Na temat obsługi takiego requestera będzie mowa przy opisie funkcji rtReqHandlerA().

RT_WaitPointer - jako parametr podajemy wartość prawdziwą (TRUE) czyli większą od zera i wtedy gdy jest włączony requester to kliknięcie myszką gdziekolwiek poza nim powoduje, że następuje zmiana wskaźnika myszki na wskaźnik uśpienia. Po wyjściu z requestera nasze okno nie będzie posiadało własnego wskaźnika (jeśli mamy własny wskaźnik należy go ustawiać po wyjściu z requestera). Zalecane jest używanie tego rozkazu bardzo często.

RTEZ_ReqTitle - jako parametr podajemy wskaźnik tekstu zakończonego zerem, który będzie używany jako nazwa requestera. Gdy nie podamy tego rozkazu to standardowo ustawiana jest nazwa "Request" do czasu gdy jest więcej niż jedna odpowiedź. Gdy odpowiedź jest jedna lub nie ma jej wcale to standardową nazwą jest "Information".

RTEZ_Flags - jako parametr podajemy długie słowo zawierające znaczniki:

EZREQF_NORETURNKEY - wyłącza klawisz RETURN jako zatwierdzenie dla odpowiedzi pozytywnej.

EZREQF_LAMIGAQUAL - zatwierdzanie requestera jest ograniczone tylko do kombinacji klawiszy AMIGA + V dla pozytywnej odpowiedzi i AMIGA + B dla odpowiedzi negatywnej.

EZREQF_CENTERTEXT - centrowanie każdej linii tekstu w oknie requestera. Bardzo przydatne do requesterów informacyjnych.

RT_DefaultResponse - jako parametr podajemy długie słowo zawierające liczbę oznaczającą, który z gadżetów ma zostać wybrany i jaka wartość ma być przekazana programowi. Rozkaz ten jest ignorowany jeżeli ustawiliśmy znacznik EZREQF_NORETURNKEY. Tekst w gadżecie, który może być zatwierdzony klawiszem RETURN jest drukowany jako wytłuszczony. Standardowo przyjmuje się wartość jeden czyli gadżet najbardziej na lewo w requesterze.

Wyjście:

Wartość - jeżeli wynik wynosi zero to został wybrany requester oznaczający negatywną odpowiedź znajdujący się najbardziej na prawo, natomiast wartości od 1 w górę oznaczają odpowiedzi od lewej strony (1 odpowiedź pozytywna). Jako wynik może też być zwrócony znacznik IDCMP, który wymusił zakończenie requestera.

---

rtReqHandlerA

Wynik = rtReqHandlerA (HandlerInfo, Signals, TagList)

A1 D0 A0

Ta funkcja powinna być wywoływana jeżeli używamy rozkazu RT_ReqHandler przy uruchamianiu requestera.

Requester, którego używamy z takim rozkazem powróci od razu gdy tylko nastąpi jego inicjalizacja i zainicjuje wskaźnik do struktury rtHandlerInfo.

Teraz należy postępować następująco:

Sprawdzamy pole DoNotWait w strukturze rtHandlerInfo i jeżeli jest ustawione na zero to musimy oczekiwać na sygnały umiesczone w polu WaitMask (oraz własne sygnały jeżeli tego chcemy). Jeżeli osiągniemy jakikolwiek z sygnałów przedstawiony w polu WaitMask (oczekiwania dokonujemy za pomocą funkcji Exec.Wait(WaitMask))lub pole DoNotWait było niezerowe to wywołujemy funkcję rtReqHandlerA() i sprawdzamy -wynik jaki zwróciła:

CALL_HANDLER - wracamy ponownie do pętli ocze-

kiwania i sprawdzania pola DoNotWait oraz wywołujemy ponownie funkcję rtReqHandlerA().

Teraz musimy przekazać sygnały jakie otrzymaliśmy z funkcji Exec.Wait(WaitMask) do funkcji rtReqHandlerA().

Wejście:

HandlerInfo - wskaźnik struktury HandlerInfo zainicjowanej przez użycie rozkazu RT_ReqHandler przy wywoływaniu requestera.

Signals - sygnały otrzymane z funkcji Exec.Wait() wywołanej z maską WaitMask ze struktury rtHandlerInfo. W przypadku gdy pole DoNotWait jest wartością większą od zera (TRUE) to ignorujemy to pole.

TagList - wskaźnik obszaru zawierającego rozkazy.

Dostępne rozkazy dla rtReqHandlerA():

RTRH_EndRequest - dostarcza rozkaz, który powoduje zamknięcie requestera. Wartością zwrócona przez requester nie będzie wartość CALL_HANDLER ale wartość zwracana przez requester. Jako parametr dla tego rozkazu podajemy warości REQ_CANCEL - powodującą zaniechanie requestera, REQ_OK powodującą zatwierdzenie requestera, natomiast w przypadku requestera typu EZRequest - podajemy numer gadżetu jaki ma być zatwierdzony (gadżet najbardziej na prawo ma numer 0 (FALSE) a najbardziej na lewo ma numer 1 (TRUE), gadżety pozostałe mają numery od 2 w górę od strony lewej do prawej.

Wyjście:

Wynik - wartość odpowiadająca wartości CALL_HANDLER jeżeli musimy jeszcze raz wywołać procedurę rtReqHandler() lub normalna wartość zwracana przez requester.

---

rtScreenToFrontSafely

rtScreenToFrontSafely (Screen)

A0

Funkcja ta ustawia dany ekran jako najbardziej widoczny (najbliższy użytkownika) jednak dopiero po sprawdzeniu, że ekran znajduje się na liście aktualnie otwartych ekranów. Ta funkcja może być użyta do ustawienia jednego ekranu do przodu po tym jak został ustawiony poprzedni ekran.

Wejście:

Screen - wskaźnik strulktury Screen dla ekranu, który ma być ustawiony z przodu obszaru wyświetlania.

---

rtSpread

rtSpread (Pozycje, Rozmiary, RozmarCałkowity, Min, Max, Ilość)

A0 A1 D0 D1 D2 D3

Funkcja ma za zadanie wyliczyć pozycje dla obiektów (na przykład gadżetów w requesterze) jeżeli podane są ich rozmiary oraz obszar, w którym mają być umieszczane. W tablicy Pozycje bedą umieszczone pozycje dla tych obiektów.

Przykład:

Ustawiamy tablicę Rozmiary na 4,6,4,2,8. Wartość RozmiarCałkowity ustawiamy na sumę tych wartości czyli 24. Pozycję początkową Min na 3 a pozycję końcową na 43 i ilość35 tych obiektów czyli 5 wpisujemy do zmiennej Ilość. Po wywołaniu tej funkcji w tablicy Pozycje będziemy mieli następujące pozycje dla tych obiektów: 3,11,19,26,31.

Wejście:

Pozycje - wskaźnik tablicy, w której będą umiesczane pozycje.

Rozmiary - wskaźnik tablicy, w której przekazujemy wymiary każdego z obiektów.

RozmiarCałkowity - suma wszystkich wartości z tablicy Rozmiary.

Min - pierwszaa pozycja jaka będzie mogła być użyta.

Max - ostatnia pozycja, pierwsza pozycja, która nie może być użyta.

Ilość - ilość obiektów czyli danych w tablicy Rozmiary.

---

rtGetVScreenSize

Wartość = rtGetVScreenSize (Screen, WskSzerokości, WskWysokości)

A0 A1 A2

Używamy tej funkcji aby zdobyć rozmiary widzianej części ekranu. Wartość zwracana przez tę funkcję może być użyta dla pionowego poszerzania. Będzie ona większa dla ekranów w trybie Interlace a mniejsza dla normalnych ekranów. Możemy używać tej wartości dla pionowego poszerzania tak aby requestery wyglądały zarówno dobrze na ekranach z włączonym trybem Interlace jak i bez niego. Aktualnie zwracane wartości to 2 dla normalnego ekranu i 4 dla trybu interlace. Nie można na tych wartościach zbytnio polegać gdyż mogą się zmienić w przyszłości. Jednak powiększenie tych wartości zostanie zachowane.

Wejście:

Screen - wskaźnik do struktury Screen dla ekranu, dla którego chcemy ustalać wymiary.

WskSzerokości - wskaźnik do komórki, do której będzie przesłana szerokość ekranu.

WskWysokości - wskaźnik do komórki, do której będzie przesłana wysokość ekranu.

Wyjście:

Wartość - wartość określająca czy ekran jest w trybie Interlace czy też nie.

Za miesiąc znowu kolejna porcja wiadomości o bibliotece ReqTools.

Marcin "Duddie" Dudar

MONKEY ISLAND II

AMIGA

Po ogromnym sukcesie, jaki odniosła gra Monkey Island i jej producent zdecydował się na wydanie drugiej części. Użytkownicy komputerów PC już od dłuższego czasu są w posiadaniu Monkey Island II, jednak wersja dla amigowców pojawiła się na rynku zaledwie kilka tygodni temu. Gra w wersji na Amigę zajmuje jedenaście dyskietek, a na zainstalowanie jej na twardym dysku trzeba mieć aż dziewięć megabajtów wolnego miejsca! Tak dużą ilość dysków z pewnością usprawiedliwia świetna grafika i animacja oraz ogromna ilość dostępnych pomieszczeń. Także efekty dźwiękowe i muzyka stoją na nieco wyższym niż w pierwszej części poziomie.

Scenariusz gry podzielony jest na cztery części. I tak pierwsza część to stworzenie lalki Largo LaGrande, druga to szukanie czterech części mapy, trzecia to ucieczka z fortecy LeChucka, a ostatnia, czwarta część to pojedynek z samym LeChuckiem. Na pewno wszyscy miłośnicy Monkey Island powinni się jak najprędzej zaopatrzyć w drugą część tego wspaniałego programu.

Teraz coś dla niecierpliwych, a mianowicie opis przejścia całej gry. Ambitnym graczom proponuję w tej chwili przerwać czytanie i zajrzeć dopiero, gdy już na prawdę nie będzie wiadomo co dalej robić. A teraz, opis:

Część pierwsza: "The Largo Embargo"

Grę rozpoczynamy na moście prowadzącym do dość oryginalnego miasta zbudowanego głównie ze statków. Z planszy tej możemy iść w dowolnym kierunku, ponieważ i tak zawsze zostaniemy złapani przez Larga. Po krótkiej rozmowie zostaniemy pozbawieni naszych pieniędzy. Nie rozpaczajmy jednak, tylko bierzmy się do roboty. Spójrzmy na znak z prawej strony mostu i po chwili (PICK UP SIGN) już mamy łopatę. Teraz idziemy w prawo i zobaczymy widok z góry na całą wyspę. Musimy pójść na bagna (SWAMP) skąd - korzystając z trumny - dopłyniemy do rezydencji kapłanki Voodoo. Na miejscu rozejrzyjmy po pierwszym pomieszczeniu. Ze stołu możemy wziąć kawałek sznurka (STRING). Teraz jeszcze spójrzmy na książki, gdy znajdziemy tytuł ASHES-TO-LIFE spróbujmy go wziąć. Po rozmowie z kapłanką Voodoo wiemy już co robić. Musimy znaleźć cztery rzeczy potrzebne do stworzenia lalki Voodoo mającej moc nad Largiem. Będą to: jego włosy, część ubrania, część zmarłego krewnego i płynu z jego ciała. Wbrew pozorom zdobycie tych rzeczy nie będzie aż takie trudne. Popłyńmy z powrotem na brzeg i przejdźmy do mapy wyspy. Teraz idziemy na cmentarz (CEMETERY) do grobów znajdujących się na górce. Tutaj , przy grobie dziadka Larga musimy zacząć kopać (USE SHOVEL ON GRAVE) i po chwili będziemy mieli jego kość (czyli część krewnego Larga). Teraz możemy już opuścić cmentarz i po ponownym wyjściu do mapy idziemy na plażę (BEACH), skąd bierzemy kijek (STICK). Teraz możemy wrócić do miasta, gdzie musimy odwiedzić Wallego-kartografa (drugie drzwi od wejścia do miasta).

Po rozmowie z Wallym weźmy kawałek papieru ze sterty kartek i monokl Willego (MONOCLE). Zauważmy, że co chwila odkłada on go na bok i przeciera oczy. Zabrać mu go musimy właśnie w tym momencie. Teraz idziemy na pierwszy statek od prawej. Czy widzisz okno położone najbardziej z lewej strony? Przez nie właśnie możemy wejść niezauważeni do kuchni, skąd szybko bierzemy nóż i niezauważeni ponownie wychodzimy przez okno. Idziemy na statek położony najdalej z lewej strony miasta . W środku, z prawej strony znajdziemy małego aligatora przywiązanego sznurkiem do kolumny. Odetnimy go przy pomocy noża. Aligator szybko wybiegnie po schodach, a zaskoczony barman popędzi za nim. Teraz możemy wziąć kawałki sera znajdujące się w misce zwierzątka. Musimy iść jeszcze do pokoju znajdującego się z lewej strony wnętrza statku. Jest to pokój Larga. Bierzemy z tamtąd perukę Larga znajdującą się na stoliku przy ścianie.

Możemy już wyjść ze statku i skierować się na górę, do rozmawiających piratów, gdzie bierzemy wiaderko. Jeden z piratów nie pozwala nam go zabrać, jednakże krótka rozmowa zmienia jego zdanie. Teraz spójrz na pudełko stojące obok. Otwórz je, a natępnie przy pomocy sznurka, patyka oraz kawałków sera zmontuj pułapkę na szczura. (USE STICK WITH BOX, USE STRING WITH STICK, USE CHEESE SQUIGGLIES WITH BOX). Po chwili zwierzaczek będzie buszował przy naszej pułapce. Teraz wystarczy tylko podnieść sznurek i mamy naszego szczura! Teraz idziemy do wejścia na pierwszym statku. Schodzimy na dół po schodach i znajdujemy się w gospodzie. Rozmawiajmy z barmanem tak długo, aż przyjdzie Largo. Po obejrzeniu animacji wiemy już, skąd zdobyć płyn z ciała Larga. Wystarczy podejść do ściany, na którą splunął i przy pomocy wziętej od kartografa kartki zdjąć ślinę (USE BLANK SHIT OF PAPER WITH DRIPPING SPIT). Możemy już wyjść z gospody. Idziemy teraz do kuchni korzystając z naszego tajnego przejścia przez okno. Tu korzystamy z naszego wcześniej złapanego szczura. Wrzucamy go do garnka z gotującą się zupą (USE RAT IN VISCIOSUES).

Ponownie wychodzimy z kuchni i wracamy do gospody. Porozmawiajmy z barmanem. Podczas romowy będziemy mieli opcję dialogową, w której prosimy o zupę. Wybierzmy ją i barman idzie do kuchni. Ponownie krótka scenka, po której kucharz wylatuje z pracy. Znowu krótka rozmowa z barmanem i my mamy posadę kucharza. Dostajemy na początek zaliczkę i idziemy do kuchni skąd natychmiast uciekamy przez okno.

Z gospody idziemy ponownie na bagna, gdzie napełniamy wiaderko błotem. Z pełnym wiaderkiem udajemy się do miasta, potem na statek, w którym znajduje się pokój Larga. Po wejściu do pokoju zamykamy drzwi i kładziemy wiadro nad nimi (USE BUCKET WITH DOOR). Gdy Largo wejdzie... Po kolejnej animacji musimy udać się za Largiem do pralni (to tam, gdzie złapaliśmy szczura). Gdy Largo załatwi już swoje sprawy możemy wrócić do jego pokoju. Po zamknięciu drzwi po drugiej ich stronie zauważymy kartkę. Jest to kwit na ubranie Larga. Bierzemy go i wracamy do pralni. Gdy damy kwit pracownikowi odda on nam ubranie Larga. W ten sposób zdobyliśmy już wszystkie części potrzebne do zbudowania lalki Voodoo, więc możemy się udać na bagna i... przy pomocy trumny do kapłanki Voodoo. Będąc w jej pokoju po kolei dajemy wszystkie zdobyte składniki, z których ona złoży dla nas lalkę Voodoo mogącą działać na Larga. Otrzymawszy ją idziemy do pokoju Larga.

Szybko, zanim zdąży nas wyrzucić musimy użyć szpilki na lalce (USE PINS ON VOODOO DOLL)... Teraz wystarczy obejrzeć kolejną animację i skończyliśmy pierwszą część gry!!!

Część druga "The Four Pieces"

Na początku idziemy na statek kapitana Dreada. Gdy damy mu monokl Wallego zgodzi się za drobną opłatą wynająć nam swój statek i zostać kapitanem. Daje nam swoją mapę, z której możemy wybrać wyspę, na którą chcemy popłynąć. Pamiętaj jednak, aby wcześniej wziąć ze statku pokarm dla papug (PARROT CHOW).

Pierwszą wyspą, na którą popłyniemy będzie PHATT. Po przybyciu złapie na strażnika bez względu na to, co zrobimy, czy też co mu powiemy. Zostajemy zaprowadzeni do gubernatora. i... wtrąceni do więzienia.

W celi podnieś materac (PICK UP MATTRESS) i weź leżący pod nim kijek (PICK UP STICK). Teraz przy jego pomocy zdobądź kość (USE STICK ON BONE) i podnieś ją z ziemi (PICK UP BONE). Zauważyłeś już chyba pieska trzymającego klucz w zębach? Spróbuj dać mu kość. Zadziałało? Podnieś teraz klucz z ziemi i otwórz drzwi celi. Przydałoby się jeszcze odzyskać wszystkie przedmioty. Podejdź do szafki i weź z niej leżące tam dwie koperty. Jedna z nich zawiera wszystkie zebrane dotąd przez ciebie rzeczy, a w drugiej znajdziesz banana oraz organki. Po wyjściu z wiezienia idź na pomost, na którym siedzi wędkarz. Porozmawiaj z nim. Możesz mu wmówić, że jesteś mistrzem wędkarstwa i założyć się z nim, że złowisz większą od niego rybę. Po skończeniu konwersacji idziemy do biblioteki (najpierw psujemy model latarni morskiej) (OPEN LIGHTHOUSE, TAKE LIGHTHOUSE LENS) i u bibliotekarki załatwiamy tymczasową kartę, z którą idziemy do katalogu książek. Musimy zabrać ze sobą trzy książki. Pierwsza z nich może być całkowicie dowolna, druga to opisy najbardziej zanych wraków. Znajduje się ona pod hasłem Disasters. Trzecia to książka o Voodoo. Jest to zestaw recept na różne czary.

Ponownie wracamy do biliotekarki i prosimy ją kolejno o trzy książki.

Teraz idziemy w lewo aż do ścieżki, którą idziemy do końca planszy. Widząc ogólny widok wyspy kierujemy się do willi gubernatora. Wchodzimy przez frontowe drzwi. Gdy podejdziemy do schodów zatrzyma strażnik. Gdy zapyta, czy nie jesteś przypadkiem Guybrushem powiedz mu, że cię musiał z kimś pomylić oraz że chyba w kuchni coś się pali - strażnik odejdzie.

Możesz już iść na góre, do pokoju gubernatora. Tu dostrzegasz, że gubernator śpi, a na pościeli leży książka. Musimy ją zdobyć, gdyż będzie nam później potrzebna.

Teraz korzystamy z książki, którą wcześniej wzięliśmy z



biblioteki (USE tutaj tytuł niepotrzebnej książki WITH FAMOUS PIRATE QUOTES) i po chwili mamy co trzeba!

AMIGA

Wracamy teraz do miasta. Z głównej ulicy wejdźmy w pierwszą przecznicę. Widzimy faceta, który wygrywa zakład i się oddala. Idziemy za nim na główną ulicę, a potem w drugą przecznicę. W magazynie chowamy się za skrzynkami. Po krótkiej animacji facet opuszcza magazyn. Warto, abyśmy także dowiedzieli się jaka liczba będzie wygraną w następnej grze. Pukamy do drzwi i prosimy o następną wygraną liczbę. Najpierw jendak będziemy musieli podać hasło. Przez okienko w drzwiach ktoś wystawi rękę i pokaże pewną ilość palców. Musimy ją zapamiętać. Za chwilę ręka pokaże inną ilość palców. Na pytanie: ile ich jest? Wtedy musimy podać tą ilość pokazaną za pierwszym razem. Gdy hasło będzie podane prawidłowo będziemy mieli kolejny wygrywający numer.

Teraz wracamy do pierwszej przecznicy i robimy zakład podając uzyskany przed chwilą numer. Wygrywamy - bierzemy więc bilet wstępu na przyjęcie na wyspie Booty, następnie powtarzamy wszystko od początku i wygrywamy kolejno pieniądze i skierowanie na urlop. Teraz możemy już wrócić na statek Dreada i popłynąć na wyspę Booty.

Na miejscu najpierw idziemy do sklepu. Kupujemy tutaj wszystko co można kupić, czyli SHIPS HORN, SIGN ABOUT PARROT, FEATHER PEN, HUB CAPS, PIRATE HATS, ELVIS PLATE, SHINY SAW, PARROT'S MIRROR. Największy problem może być z lustrem (PARROT'S MIRROR), gdyż sprzedawca nie chce go nam sprzedać. Aby je dostać należy najpierw kupić SIGN ABOUT PARROT. Teraz na haku pozostalym po tabliczce powiesimy zabraną ze staku Dreada torebkę (USE EMPTY BAG OF PARROT CHOW ON HOOK). Papuga zacznie się teraz przyglądać torebce i będziemy mogli kupić lustro. Gdy kupisz już wszytko porozmawiaj o leżącej na ladzie mapie. Niestety nie będziemy mogli jej kupić, ale sprzedawca będze skłonny ją wymienić za głowę małpy z zaginionego statku Mad Monkey. Figurą jednak zajmiemy się później. Po wyjściu z antykwariatu idziemy do sklepu z kostiumami, w którym - po pokazaniu biletu wstępu na przyjęcie otrzymamy nasz kostium. Teraz jeszcze weźmiemy od stojącej obok Kate ulotkę informacyjną jej firmy i możemy iść na przyjęcie. Kierujemy się w prawo do samego końca ekranu i wychodzimy na mapę wyspy. Oczywiście idziemy do willi gubernatorki. Po drodze spotkasz strażniczkę, która każe się przebrać w kostium, po chwili wchodzimy na teren przyjęcia i po krótkiej przeprawie przez las docieramy do willi. Najpierw idziemy za willę i tam przewracamy kosze na śmieci (PUSH GARBAGE CANS) powodując dużo hałasu. Z kuchni wybiegnie kucharz i zacznie nas gonić. Musimy obiec willę dookoła, szybko wejść do kuchni i zabrać jedną z ryb leżących w garnku. Teraz możemy już wejść do willi. Z wiszącego w głównym holu obrazu weźmiemy kawałek mapy. Wychodzimy z willi, ale niestety złapani przez ogrodnika zostajemy odprowadzeni do pokoju gubernatorki. Tam, po dość krótkiej konwersacji nasza mapa zostaje wyrzucona przez okno. Możemy już wyjść z willi. Spróbujmy podnieść leżącą obok mapę. Niestety mapa odlatuje coraz dalej, aż w końcu ląduje na klifach po drugiej stronie wyspy. Niestety nic w tej sprawie w tej chwili nie możemy poradzić. Wracamy do willi, do pokoju Elaine. Nad jej łóżkiem wisi wiosło (OAR), które musimy wziąć. Przy wyjściu z willi trzeba jeszcze zabrać ze sobą psa. Nie przejmujmy się jego romiarami, gdyż bez problemu zmieści się w kieszeni. Teraz na statek Dreada i na wyspe Phatt! Wędkarzowi dajemy rybę, za co dostaniemy jego wędkę (wygraliśmy zakład). Z nią wracamy na wyspę Booty i idziemy na

klify. Tam, korzystając z wędki będziemy próbowali zdobyć nasz kawałek mapy. Niestety, gdy już prawie go mamy w ręku zostaje on porwany przez ptaka.

Idziemy za ptakiem do wielkiego drzewa. Widzisz te małe dziurki w jego pniu? Trzeba teraz w pierwszą z nich włorzyć wiosło (USE OAR WITH HOLE). Wchodzimy na korzeń, na pierwszy stopień, a następnie na nasze wiosło i... No tak, straciliśmy przytomność.

Po kolejnej animacji bierzemy złamane wiosło i jedziemy na wyspę Scab. Idziemy oczywiście do cieśli (pierwsze drzwi), u którego naprawiamy wiosło. Teraz jeszcze damy Wallemu coś, co przypomina jego monokl (GIVE LIGHTHOUSE LENS TO WALLY). W końcu idziemy do naszej zaprzyjaźnionej gospody.

Zamów wszystkie trzy kolorowe drinki; oprócz nich dostaniesz za darmo słomkę (CRAZY STRAW). Poproś jeszcze o prawie-grog, ale barman powie, że cały został wykupiony przez Kate.

Teraz zainteresujemy się małpą. Na pianinie stoi metronom, do którego wkładamy banan. Możemy wziąć ze sobą zahipnotyzowaną małpę. Wychodzimy z gospody i idziemy na górę, do pralni. Podejdź do pirata z drewnianą nogą i przy pomocy piły utnij ją. Będzie krzyczał - za chwilę przybiegnie tu cieśla. Możemy teraz iść do jego warsztatu, skąd bierzemy młotek oraz gwoździe. Kierujemy się teraz na wyspę Booty, a na niej do sklepu Stana. Po krótkiej rozmowie wejdzie do trumny, aby zademonstrować jej jakość. Zamknij go szybko przy użyciu gwoździ oraz młotka. Ze sklepu bierzemy klucz do krypty (CRYPT KEY). Po wyjściu idziemy dalej w prawo na konkurs plucia. Tutaj musimy użyć rogu (SHIP HORN) i gdy sędzia na moment opuści miejsce konkursu szybko poprzestawiać flagi (PICK UP FLAG). Mieszając wzięte z baru drinki żółty z niebieskim otrzymasz zielony. Wypij go używając słomki. Teraz możesz przystąpić do konkursu. Zwróć uwagę na kierunek wiatru, gdy będzie sprzyjający musisz kliknąć na komendę PTOOIE. Po chwili nagroda jest nasza.

Idziemy ponownie do sklepu z antykami, gdzie sprzedajemy zdobytą przed chwilą plakietkę (właściciela musimy przekonać o jej wartości).

Mając pieniądze wyjdźmy ze sklepu i zajrzyjmy do książki o zaginionych wrakach - dowiemy się o położeniu statku Mad Monkey. Następnie idziemy do Kate, od której wynajmujemy statek. Miejsce do którego chcemy popłynąć znajdujemy na mapie Dreada. Po dopłynięciu do celu skaczemy do wody i płyniemy do samego dna, gdzie znajdujemy zaginiony statek wraz z wizerunkiem małpy. Bierzemy głowę i podchodzimy do liny. Jedno szarpnięcie i już po chwili jesteśmy na statku, a po krótkiej podróży ponownie na wyspie.

Idziemy oczywiście do sklepu, gdzie dajemy sprzedawcy głowę małpy. W zamian za nią otrzymujemy pierwszy fragment mapy. Teraz idziemy do wielkiego drzewa, ponownie używamy wiosła. Stojąc na nim wyjmij z pnia pierwszy stopień i włórz go do kolejnej dziury. Już chyba wiemy o co chodzi, a i nasz bohater po chwili się domyśli.

Teraz idziemy do domku położonego na górze z lewej strony. Bierzemy teleskop i wracamy do pierwszego domku. Widzimy dość dużą ilość różnych map, tak więc mamy kłopot ze znalezieniem tej jednej potrzebnej mapy. Ale od czego mamy pieska? On napewno znajdzie własność Elaine (USE DOGGIE ON PILE OF MAPS). Po chwili mamy już drugi fragment mapy. Wracamy ponownie na wyspę Scab, idziemy na cmentarz. Tu - przy pomocy klucza wziętego od Stana - otwieramy kryptę, w której stoi kilka trumien; jedną z nich możemy otworzyć. W środku znajduje się garstka popiołów, bierzemy je ze sobą. Idziemy teraz do kapłanki Voodoo. Dajemy jej popioły, a następnie gdy powie, że nie pamięta recepty , odpowiednią książkę.

Mając substancję przywracającą do życia wracamy do krypty i używamy jej na pozostałych w trumnie popiołach. Kucharz ożywa i po krótkiej rozmowie daje nam klucz. Idziemy teraz na plażę, tam podchodzimy do sotjącego na niej domku i korzystając z klucza otwieramy drzwi. W domku gasimy kuchenkę. Wracamy na cmentarz i ponownie ożywiamy kucharza. W zamian za przysługę daje on nam trzecią cześć mapy!

Płyniemy teraz na wyspę Phatt. Na miejscu podchodzimy do plakatu, na którym jest informacja o nas i naklejamy nań zdjęcie Kate (to z ulotki). Wracamy na nasz statek. Po skończonej animacji idziemy do więzienia, skąd bierzemy kopertę Kate, a z niej wyjmujemy butelkę prawie-grogu. Wychodzimy teraz z miasta na mapę wyspy i idziemy do wodospadów (WATERFALLS). Tutaj wchodzimy na samą górę i gdy znajdziemy pompę użyjemy małpy (to gra słów MONKEYwrench to po angielsku klucz francuski). W ten prosty sposób wyłączyliśmy wodospad i odnaleźliśmy ukryte przejście. Wchodzimy w tunel i po krótkim spacerze znajdujemy się na małej wysepce. Musimy iść na górę aż do domku stojącego na szczycie, na miejscu otwieramy drzwi i wchodzimy do środka. Spotkamy tu faceta, który zmusi nas do zmierzenia się z nim w piciu grogu. Przyniesie nam pełny kubek tego napoju i pójdzie po następny, dla siebie. W tym czasie musimy wylać grog do doniczki stojącej obok i napełnić kubek naszym prawie-grogiem. Gospodarz po chwili wraca ze swoją porcją i konkurs się zaczyna. Wypijmy naszą porcję, nasz gospodarz pije swój grog, ale ze znacznie gorszym skutkiem. W ten sposób wygraliśmy konkurs! Rozejrzyjmy się teraz po wnętrzu. Z pewnością warto by wstawić lustro w wyraźnie nań przeznaczone miejsce. Teraz jeszcze musimy otworzyć okno znajdujące się na prawo od drzwi i możemy wyjść z domku.

Podejdź do statuetki stojącej na zewnątrz i włóż jej do ręki teleskop (USE TELESCOPE WITH UGLY STATUE). Po chwili promień światła wskaże nam jedną z cegieł w ścianie domku. Musimy ją zapamiętać. Ponownie wchodzimy do domku, gdzie naciskamy na zaznaczoną przez promień cegłę. Spadamy do piwnicy, w której znajduje się szkielet trzymający czwartą część mapy, którą oczywiście natychmiast bierzemy. Wracamy teraz na statek Dreada i płyniemy na wyspę Scab. Na wyspie idziemy do kartografa, któremu dajemy kolejne części mapy. Obiecuje on nam, iż je poskłada, ale w zamian za to musimy iść do kapłanki Voodoo po bombę miłości. Zostawiamy Wallego i idziemy do kapłanki. Po rozmowie i wysłuchaniu jej wizji dostajemy torbę z przygotowanymi dla Wallego rzeczami. Wracamy do miasta, do pracowni kartograficznej Willego i niestety okazuje się, że wizja kapłanki w pełni się sprawdziła. Zrozpaczeni wracamy na bagna, gdzie znajdujemy dość sporych romiarów pudełko (CRATE) zaadresowane do LeChucka. Otwieramy je i wchodzimy do środka.

Tak się właśnie kończy druga część naszej gry.

Ciąg dalszy nastąpi...

Jarosław "Jarri" Horodecki